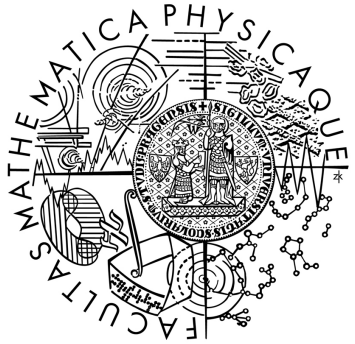
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Veronika Půlpánová

Homomorphic encryption and coding theory

Department of algebra

Supervisor of the master thesis: RNDr. Michal Hojsík Ph.D.

Study programme: Mathematics

Specialization: Mathematical Methods of Information Security

Prague 2012

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Název práce: Homomorphic encryption and coding theory

Autor: Veronika Půlpánová

Katedra: Katedra algebry

Vedoucí diplomové práce: RNDr. Michal Hojsík, Ph.D., Katedra algebry

Abstrakt: V dnešní době se většina výzkumu v plně homomorfním šifrování opírá o teorii mříží. Tato práce se zabývá alternativními přístupy k homomorfnímu šifrování. Nejdříve si ukážeme homomorfní kryptosystém, založený na teorii kódů, navržený Armknechtem a kol. a popíšeme jeho vlastnosti. Dále prezentujeme kryptosystémy ze skupiny známé jako Polly Cracker a poukážeme si některé jejich problematické aspekty. Hlavním přínosem této práce je design nového symetrického plně homomorfního kryptosystému, který je založen na Polly Crackeru. Navrhujeme nový přístup k řešení problému vysoké výpočetní složitosti u jednoduchých Polly Cracker systémů. Design používá Gröbnerovy báze pro generování zero-dimensional ideals polynomiálních okruhů nad konečnými tělesy, jejichž faktorokruhy dále používá jako okruhy šifrových textů. Gröbnerovy báze dávají těmto okruhům snadno algoritmizovatelnou multiplikativní strukturu, vhodnou pro plně homomorfní šifrování.

Klíčová slova: Plně homomorfní šifrování, Polly Cracker, teorie kódů, zero-dimensional ideals

Title: Homomorphic encryption and coding theory

Author: Veronika Půlpánová

Department: Department of algebra

Supervisor: RNDr. Michal Hojsík, Ph.D., Department of algebra

Abstract: The current mainstream in fully homomorphic encryption is the approach that uses the theory of lattices. The thesis explores alternative approaches to homomorphic encryption. First we present a code-based homomorphic encryption scheme by Armknecht et. al. and study its properties. Then we describe the family of cryptosystems commonly known as Polly Cracker and identify its problematic aspects. The main contribution of this thesis is the design of a new fully homomorphic symmetric encryption scheme based on Polly Cracker. It proposes a new approach to overcoming the complexity of the simple Polly Cracker - based cryptosystems. It uses Gröbner bases to generate zero-dimensional ideals of polynomial rings over finite fields whose factor rings are then used as the rings of ciphertexts. Gröbner bases equip these rings with a multiplicative structure that is easily algorithmized, thus providing an environment for a fully homomorphic cryptosystem.

Keywords: Fully homomorphic encryption, Polly Cracker, coding theory, zero-dimensional ideals

Contents

Introduction

As it becomes more and more common to outsource calculations with data, the demand for a secure, effective, fully homomorphic encryption scheme rises. Such scheme would allow for evaluation of multivariate polynomials over ciphertexts. A customer could encrypt his data, send it for calculations and receive the encrypted result. Decryption would reveal the result of the same calculations performed on the plaintext data. No one but the customer, who knows the secret key could gain knowledge about either plaintext data or the calculated result.

The first chapter describes the concept of homomorphic encryption as an important cryptographic primitive with a variety of practical use. The cryptosystems that are homomorphic with respect to either addition or multiplication have been well known since the appearance of RSA cryptosystem. They are used in protocols for distance electronic voting, e.g. in [FOO92], electronic bidding, e.g. the Dutch crop auctions [BCD+08] and many other protocols.

A fully homomorphic cryptosystem, called Polly Cracker has been presented in 1994 in [FK94] by Neal Koblitz and Michael Fellows. We describe this scheme in Chapter 4. As we show, the parameters in this scheme cannot be set up in such way that it would be both effective and secure.

Certainly the most promising fully homomorphic scheme so far has been presented by Craig Gentry in 2009 in [Gen09]. The scheme is loosely based on Polly Cracker. It uses the theory of lattices and many sophisticated tricks such as self-evaluating circuits. Detailed description of this scheme is out of the scope of this thesis. Let us just point out, that it allows for an unlimited number of multiplications, does not have any bound on the number of plaintexts encrypted with the same key and its security is based on certain worst-case problems over ideal lattices and the sparse subset sum problem. The downside of the scheme is the size of the keys (in gigabytes) and the high complexity of operations with the ciphertexts, which make it impractical for commercial use at this stage. However, the scheme is still under development and is becoming more and more practical with each paper published on the topic.

Another noteworthy paper on homomorphic encryption [AAPS11] has been published by Armknecht et. al. in 2011. It uses a completely different approach to the problem of homomorphic encryption. Instead of ideals of multivariate polynomial rings it works with linear codes. It uses their natural property of being additive. We study this scheme closely in Chapter 2. The security of the scheme is based on the assumed hardness of decoding in a random code. Its limitation is a tight bound on the number of plaintexts that can be encrypted with one key and a bound on the number of multiplications performed on the ciphertexts. This bound is arbitrary, but the complexity of the operations on ciphertexts rises quickly with it. We explore some ways of overcoming this limitation in Section 2.3, but we do not succeed.

Chapter 3 reviews basic definitions and propositions about Gröbner bases and zero-dimensional ideals.

In Chapter 5 we present a new symmetric encryption scheme SymPC-version 1 that is fully homomorphic. It is based on Polly Cracker, but the Gröbner bases play a totally different role here. Instead of being a threat, as an attacker, who

computes some Gröbner basis could break the system with it, in this scheme we take advantage of Gröbner bases to significantly reduce the complexity of all Setup, Encryption, Decryption and operations with the ciphertexts. Complexity of these is treated in Section 5.1.

In Chapter 6 we design two attacks at SymPC1. The following Chapter 7 suggests parameter settings so that these two attacks may be avoided and further explores the security of SymPC1.

We conclude the thesis by Chapter 8, where we propose a second version SymPC2, which has a lower complexity of some of the functions and is optimized, so that even a weaker setting of parameters avoids the two attacks from Chapter 6. We conjecture, that the changes made to the algorithm between versions SymPC1 and SymPC2 do not affect the security of the cryptosystem.

1. Preliminaries

We begin with a definition of a cryptosystem as presented by Stinson in [Sti95].

Definition 1. *A cryptosystem is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:*

1. *$\mathcal{P}$ is a finite set of possible plaintexts*
2. *$\mathcal{C}$ is a finite set of possible ciphertexts*
3. *$\mathcal{K}$, the keyspace, is a finite set of possible keys*
4. *For each $K \in \mathcal{K}$, there is an encryption rule $e_K \in \mathcal{E}$ and a corresponding decryption rule $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \longrightarrow \mathcal{C}$ and $d_K : \mathcal{C} \longrightarrow \mathcal{P}$ are functions such that $d_K(e_K(x)) = x$ for every plaintext $x \in \mathcal{P}$*

Note that the Condition 4 implies that e_K is injective for all $K \in \mathcal{K}$. Let us take $x_1, x_2 \in \mathcal{P}$ and $K \in \mathcal{K}$ such that $e_K(x_1) = e_K(x_2) = y$. Then Condition 4 says that $d_K(y) = x_1$ and $d_K(y) = x_2$, therefore $x_1 = x_2$ and e_K is injective for all K.

Definition 2 (Probabilistic Cryptosystem)**.** *Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem. If for all $K \in \mathcal{K}$, the function $e_K : \mathcal{P} \longrightarrow \mathcal{C}$ is non-deterministic, we call such cryptosystem probabilistic. In other words, there are many different ways to encrypt a message m with the key K and $e_K(m)$ is a subset of $\mathcal{C}$.*

Sometimes we shall use $e_K(m)$ to denote a particular random encryption of m and sometimes to denote the set of all possible encryptions. In most cases it will not matter and in others the meaning should be clear from the context.

Clearly, it the case of probabilistic cryptosystems, the function d_K cannot be injective. It may happen, that $e_K(\mathcal{P}) \subsetneq \mathcal{C}$. Then there are such $\mathbf{w} \in \mathcal{C}$, that $d_K(\mathbf{w}) = m \in \mathcal{P}$, but $\mathbf{w} \notin e_K(m)$. Examples of such cryptosystems will be shown later.

Definition 3 (Homomorphic Cryptosystem)**.** *Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem. If $\mathcal{P} = \mathcal{P}(\alpha_1, \ldots, \alpha_n)$ and $\mathcal{C} = \mathcal{C}(\beta, \ldots, \beta_n)$ are algebras and if for all $K \in \mathcal{K}$, e_K is a homomorphism of algebras $\mathcal{P}$ and $\mathcal{C}$, we say that cryptosystem $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is homomorphic with respect to $\alpha_1, \ldots, \alpha_n$.*

Example 4 (RSA cryptosystem)**.** *Let p, q be distinct primes, $n = p \cdot q$ and let $\mathcal{K} = \{K = (e, d) \mid e, d \in \mathbb{Z}^*_{\varphi(n)},\ d = e^{-1} \mod \varphi(n)\}$. Let*

$$\begin{array}{rrcl} e_K: & \mathbb{Z}_n(\cdot, 1) & \longrightarrow & \mathbb{Z}_n(\cdot, 1) \\ & x & \longmapsto & x^e \mod n \\ d_K: & \mathbb{Z}_n(\cdot, 1) & \longrightarrow & \mathbb{Z}_n(\cdot, 1) \\ & x & \longmapsto & x^d \mod n \end{array}$$

Then $(\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{K}, \{e_K \mid K \in \mathcal{K}\}, \{d_K \mid K \in \mathcal{K}\})$ is called RSA cryptosystem.

The function e_K is a homomorphism of monoids:

$$e_K(x_1) = x_1^e,\ e_K(x_2) = x_2^e$$
$$e_K(x_1 \cdot x_2) = (x_1 \cdot x_2)^e = x_1^e \cdot x_2^e = e_K(x_1) \cdot e_K(x_2)\ .$$

Therefore RSA is a multiplicatively homomorphic cryptosystem.

Our goal is to construct an encryption scheme that would be homomorphic with respect to addition and multiplication, i.e. for all $m_1, m_2 \in \mathcal{P}$ this would allow us to compute $e(m_1 \cdot m_2)$ and $e(m_1 + m_2)$ from the knowledge of $e(m_1)$ and $e(m_2)$. Such schemes are often referred to as fully homomorphic.

1.1 Remarks on the notation

In R^n, where R is a commutative ring and $n \in \mathbb{N}$, the addition and multiplication is always meant componentwise. We shall use this notation throughout the whole thesis and it will always be denoted as regular $+$ and $\cdot$.

$$\begin{array}{rrcl} + : & R^n \times R^n & \longrightarrow & R^n \\ & (r_1, \ldots, r_n) + (r'_1, \ldots, r'_n) & \longmapsto & (r_1 + r'_1, \ldots, r_n + r'_n) \\ \cdot : & R^n \times R^n & \longrightarrow & R^n \\ & (r_1, \ldots, r_n) \cdot (r'_1, \ldots, r'_n) & \longmapsto & (r_1 \cdot r'_1, \ldots, r_n \cdot r'_n) \end{array}$$

The $+$ in "$r_i + r'_i$" and $\cdot$ in "$r_i \cdot r'_i$" mean addition and multiplication in the ring R. We shall also use the notation

$$\prod_{j=1}^{m} r^{(j)} = \left(\prod_{j=1}^{m} r_1^{(j)}, \ldots, \prod_{j=1}^{m} r_n^{(j)} \right)\ .$$

Let A be a finite set. Then $a \leftarrow A$ denotes: Choose a from A uniformly at random.

Throughout the thesis, $\mathbb{F}$ shall denote a finite field.

1.2 Lagrange interpolation

In the following section we define two functions Lag and Ev. Let $x_1, \ldots, x_n \in \mathbb{F}$ be n distinct points in $\mathbb{F}$, $\mathbf{x} := (x_1, \ldots, x_n) \in \mathbb{F}^n$. The function Ev evaluates a polynomial in the n points of $\mathbf{x}$ to get a vector in $\mathbb{F}^n$. The function Lag interpolates a vector in the n points of $\mathbf{x}$ to get a polynomial in $\mathbb{F}[x]$ of a degree less than n. We denote $\mathcal{L} := \{p \in \mathbb{F}[x] \mid \deg(p) < n\}$.

Definition 5. *Let $\mathbb{F}$ be a finite field, $|\mathbb{F}| \geq n$ and let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ such that all x_i are distinct. We define mappings*

$$\begin{array}{rrcl} \mathrm{Lag}_{\mathbf{x}}: & \mathbb{F}^n & \longrightarrow & \mathbb{F}[x] \\ & \mathbf{w}=(w_1,\ldots,w_n) & \longmapsto & \sum_{i=1}^{n} w_i \prod_{j=1,j\neq i} \frac{x-x_j}{x_i-x_j} \end{array}$$

$$\begin{array}{rrcl} \mathrm{Ev}_{\mathbf{x}}: & \mathbb{F}[x] & \longrightarrow & \mathbb{F}^n \\ & p & \longmapsto & (p(x_1),\ldots,p(x_n)) \end{array}$$

Proposition 6. *The function* $\mathrm{Ev}_{\mathbf{x}}$ *is a bijection of* $\mathcal{L}$ *to* $\mathbb{F}^n$*. The function* $\mathrm{Lag}_{\mathbf{x}}$ *is its inverse, i.e.* $\mathrm{Ev}_{\mathbf{x}} \circ \mathrm{Lag}_{\mathbf{x}} = Id_{\mathbb{F}^n}$*.*

Proof. **Injectivity:** Let $p, q \in \mathbb{F}[x]$, such that $p(x_i) = q(x_i)$ for all $i \in \{1,\ldots,n\}$, $\deg(p) < n$ and $\deg(q) < n$. Then $(p-q)(x_i) = 0$ for all $i \in \{1,\ldots,n\}$ and $\deg(p-q) < n$. The polynomial $p-q$ has at least n roots, which is more than its degree, so $p - q \equiv 0 \in \mathbb{F}[x]$ and $p = q$.

The size of $\mathbb{F}^n$ is $|\mathbb{F}|^n$ and the number of polynomials of a degree less than n over $\mathbb{F}$ is also $|\mathbb{F}|^n$. $\mathrm{Ev}_{\mathbf{x}}$ is an injective mapping of two sets of the same size, so it has to be bijective.

We need to prove that for all $\mathbf{w} \in \mathbb{F}^n$ it holds that $\mathrm{Ev}_{\mathbf{x}}\left(\mathrm{Lag}_{\mathbf{x}}(\mathbf{w})\right) = \mathbf{w}$, i.e.

$$\forall l \in \{1,\ldots,n\} : \left(\mathrm{Lag}_{\mathbf{x}}(\mathbf{w})\right)(x_l) = w_l \ .$$

We have

$$\left(\mathrm{Lag}_{\mathbf{x}}(\mathbf{w})\right)(x_l) = \sum_{i=1}^{n} w_i \prod_{j=1,j\neq i} \frac{x-x_j}{x_i-x_j}(x_l) = \sum_{i=1}^{n} w_i \prod_{j=1,j\neq i} \frac{x_l-x_j}{x_i-x_j}$$

For $i \neq l$ we have $\prod_{j=1,j\neq i} \frac{x_l-x_j}{x_i-x_j} = 0$. Therefore

$$\left(\mathrm{Lag}_{\mathbf{x}}(\mathbf{w})\right)(x_l) = w_l \prod_{j=1,j\neq l} \frac{x_l-x_j}{x_l-x_j} = w_l \cdot 1 = w_l$$

and $\mathrm{Ev}_{\mathbf{x}} \circ \mathrm{Lag}_{\mathbf{x}}$ is an identity on $\mathbb{F}^n$. □

Proposition 7. *For the composition of the functions* $\mathrm{Lag}_{\mathbf{x}}$ *and* $\mathrm{Ev}_{\mathbf{x}}$ *it holds*

$$\begin{array}{rrcl} \mathrm{Lag}_{\mathbf{x}} \circ \mathrm{Ev}_{\mathbf{x}}: & \mathbb{F}[x] & \longrightarrow & \mathcal{L} \\ & p & \longmapsto & p \mod \prod_{i=1}^{n}(x-x_i) \ . \end{array}$$

Proof. From the previous proposition we can see that $\mathrm{Lag}_{\mathbf{x}} \circ \mathrm{Ev}_{\mathbf{x}} = Id_{\mathcal{L}}$, i.e. for any polynomial $p \in \mathbb{F}[x]$ of a degree less than n we have

$$\mathrm{Lag}_{\mathbf{x}} \circ \mathrm{Ev}_{\mathbf{x}}(p) = p = p \mod \prod_{i=1}^{n}(x-x_i) \ .$$

Now let $\deg(p) \geq n$. Set $p' = p \mod \prod_{i=1}^{n}(x-x_i)$ and $q \in \mathbb{F}[x]$, such that $p = p' + q \cdot \prod_{i=1}^{n}(x-x_i)$. Then

$$\begin{aligned} \mathrm{Ev}_{\mathbf{x}}(p) &= \mathrm{Ev}_{\mathbf{x}}\left(p' + q \cdot \prod_{i=1}^{n}(x-x_i)\right) \\ &= \mathrm{Ev}_{\mathbf{x}}(p') + \mathrm{Ev}_{\mathbf{x}}\left(q \cdot \prod_{i=1}^{n}(x-x_i)\right) \\ &= \mathrm{Ev}_{\mathbf{x}}(p') \ . \end{aligned}$$

We have

$$\mathrm{Lag}_{\mathbf{x}} \circ \mathrm{Ev}_{\mathbf{x}}(p) = \mathrm{Lag}_{\mathbf{x}} \circ \mathrm{Ev}_{\mathbf{x}}\left(p'\right) = p' \ ,$$

which concludes the proof. □

2. Homomorphic encryption schemes from linear codes

The symmetric encryption scheme presented by Armknecht et al. in [AAPS11] (to be described in detail in Section 2.2) is rather inspiring than useful. It uses the natural property of linear codes being additive, but struggles with the multiplicativity, as we will see towards the end of this chapter.

Suppose we have $m_1, m_2, m_3 \in \mathbb{F}$, we would like to calculate the product $m_1 \cdot m_2 \cdot m_3 \in \mathbb{F}$ and we want to outsource this calculation. However, we do not want the company, who performs the calculation for us, to know neither m_1, m_2, m_3 nor $m_1 \cdot m_2 \cdot m_3$. We can also tolerate if the company has to do some extra calculations. In the ideal scenario, we would like to have a multiplicatively homomorphic scheme. For now, we construct a scheme with a weaker property. The following definition is a generalization of a definition from [AAPS11].

Definition 8 (μ-multiplicative cryptosystem)**.** *Let $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cryptosystem, $\mu \in \mathbb{N}$. If for every $K \in \mathcal{K}$ and every $\mathbf{w}^{(1)}, ..., \mathbf{w}^{(l)} \in e_K(\mathcal{P})$, the "fresh encryptions", where $l \leq \mu$, it holds that $\prod_{j=1}^{l} \mathbf{w}^{(j)} \in \mathcal{C}$ and*

$$d_K\left(\prod_{j=1}^{l} \mathbf{w}^{(j)}\right) = \prod_{j=1}^{l} d_K(\mathbf{w}^{(j)}) \ ,$$

we call such cryptosystem μ-multiplicative.

In the definition we used the term "fresh encryption". An element $\mathbf{w} \in \mathcal{C}$ is a fresh encryption, if there exists such $m \in \mathcal{P}$, that $\mathbf{w} \in e_K(m)$. Note, that in general, this is not equivalent to the fact, that there exists such $m \in \mathcal{P}$, that $d_K(\mathbf{w}) = m$.

Recall, that we assume that $e_K(P) \subseteq \mathcal{C}$. In the above definition we assume only that $\prod_{j=1}^{l} \mathbf{w}^{(j)} \in \mathcal{C}$, and not necessarily in $e_K(P)$. In other words, $\prod_{j=1}^{l} \mathbf{w}^{(j)}$ might not be one of possible encryptions of $\prod_{j=1}^{l} m_j$, however, the definition of μ-multiplicativity requires that it will be decrypted to $\prod_{j=1}^{l} m_j$.

In the scheme proposed by Armknecht et al.[AAPS11] the multiplicativity is achieved by a trick with multiplicative codes. Let us explain this concept on a simplified example.

2.1 Example of 3-multiplicative scheme

In this section we describe a simple probabilistic 3-multiplicative cryptosystem. Let $\mathbb{F}$ be a finite field $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$. We fix the secret key $K = (\mathbf{x}, y)$, where $y \in \mathbb{F}$ and $\mathbf{x} = (x_1, x_2, x_3, x_4) \in \mathbb{F}^4$ in such way that all x_i and y are distinct. We use the set of plaintexts $\mathcal{P} = \mathbb{F}$ and the set of ciphertexts $\mathcal{C} \subseteq \mathbb{F}^4$. The encoding shall be denoted Enc_K.

$$\mathbb{F} \xrightarrow{\quad \mathrm{Enc}_K \quad} \mathbb{F}^4$$

It is done in two steps - Expression and Evaluation. The two functions are denoted Ex_y and $\mathrm{Ev}_{\mathbf{x}}$. Ex_y depends on the secret element $y \in \mathbb{F}$ and $\mathrm{Ev}_{\mathbf{x}}$ depends on the secret vector $\mathbf{x} \in \mathbb{F}^4$.

$$\begin{array}{ccccc} \mathbb{F} & \xrightarrow{\mathrm{Ex}_y} & \mathbb{F}[x] & \xrightarrow{\mathrm{Ev}_{\mathbf{x}}} & \mathbb{F}^4 \\ m & \longmapsto & p_m & \longmapsto & \mathbf{w} \end{array}$$

The function Expression works as follows: for a message $m \in \mathbb{F}$ take a random polynomial $p^* \in \mathbb{F}[x]$ of a degree at most 1 and set

$$\mathrm{Ex}_y(m) = p_m := p^* - p^*(y) + m \in \mathbb{F}[x] \ .$$

We see that the function assigns to a message m a random polynomial of degree at most one, such that $p_m(y) = m$ and it is probabilistic.

The function Evaluation takes a polynomial $p \in \mathbb{F}[x]$ and evaluates it in the four points of $\mathbf{x} \in \mathbb{F}^4$.

$$\mathrm{Ev}_{\mathbf{x}}(p) = \mathbf{w} := (p(x_1), \ldots, p(x_4)) \ \in \mathbb{F}^4$$

Our encoding Enc_K is a probabilistic function and it is a composition of the probabilistic function Ex_y and a deterministic function $\mathrm{Ev}_{\mathbf{x}}$.

$$\begin{array}{rccccc} \mathrm{Enc}_K: & \mathbb{F} & \xrightarrow{\mathrm{Ex}_y} & \mathbb{F}[x] & \xrightarrow{\mathrm{Ev}_{\mathbf{x}}} & \mathcal{C} \\ & m & \longmapsto & p_m & \longmapsto & \mathbf{w} \end{array}$$

The decoding is done as follows:

$$\begin{array}{rccccc} \mathrm{Dec}_K: & \mathbb{F}^4 & \xrightarrow{\mathrm{Lag}_{\mathbf{x}}} & \mathbb{F}[x] & \xrightarrow{\mathrm{Ev}_y} & \mathbb{F} \\ & \mathbf{w} & \longmapsto & p_{\mathbf{w}} & \longmapsto & p_{\mathbf{w}}(y) \end{array}$$

As we have seen in Section 1.2, the function $\mathrm{Lag}_{\mathbf{x}}$ is the inverse to $\mathrm{Ev}_{\mathbf{x}}$, as long as the random polynomial p_m was chosen in such way, that its degree was strictly less than four. Correctness of the scheme is quite straightforward:

$$\begin{aligned} \mathrm{Dec}_K\left(\mathrm{Enc}_K(m)\right) &= \mathrm{Dec}_K\left(\mathrm{Ev}_{\mathbf{x}}\left(\mathrm{Ex}_y(m)\right)\right) = \mathrm{Ev}_y(\mathrm{Lag}_{\mathbf{x}}(\mathrm{Ev}_{\mathbf{x}}\left(\mathrm{Ex}_y(m)\right)) \\ &= \mathrm{Ev}_y(\mathrm{Ex}_y(m)) = (\mathrm{Ex}_y(m))(y) = (p^* - p^*(y) + m)(y) = m \end{aligned}$$

Now let us take $m_1, m_2, m_3 \in \mathbb{F}$ and $p_1, p_2, p_3 \in \mathbb{F}[x]$, their respective possible images by Ex_y. We set $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)} \in \mathbb{F}^4$ the evaluations of p_1, p_2, p_3 respectively, $\mathbf{w}^{(j)} = (w_1^{(j)}, \ldots, w_4^{(j)})$. Then $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}$ are possible encodings of m_1, m_2, m_3 respectively and $\mathrm{Dec}_K\left(\mathbf{w}^{(j)}\right) = m_j$ for $j \in \{1, 2, 3\}$.

$$\begin{array}{ccccc} m_1 & \xmapsto{\mathrm{Ex}_y} & p_1 & \xmapsto{\mathrm{Ev}_{\mathbf{x}}} & \mathbf{w}^{(1)} \\ m_2 & \xmapsto{\mathrm{Ex}_y} & p_2 & \xmapsto{\mathrm{Ev}_{\mathbf{x}}} & \mathbf{w}^{(2)} \\ m_3 & \xmapsto{\mathrm{Ex}_y} & p_3 & \xmapsto{\mathrm{Ev}_{\mathbf{x}}} & \mathbf{w}^{(3)} \end{array}$$

Now we show that this cryptosystem is 3-multiplicative. In other words, if we take three encodings of messages, multiply them through and then we decode them, we get the exact same result as if we first decoded them and then we multiplied them through.

$$\mathrm{Dec}_K\left(\mathbf{w}^{(1)} \cdot \mathbf{w}^{(2)} \cdot \mathbf{w}^{(3)}\right) = \mathrm{Dec}_K\left(\mathbf{w}^{(1)}\right) \cdot \mathrm{Dec}_K\left(\mathbf{w}^{(2)}\right) \cdot \mathrm{Dec}_K\left(\mathbf{w}^{(3)}\right) = m_1 \cdot m_2 \cdot m_3.$$

We set $\mathbf{w} = (w_1, \ldots, w_4) := \mathbf{w}^{(1)} \cdot \mathbf{w}^{(2)} \cdot \mathbf{w}^{(3)} \in \mathbb{F}^4$. We are going to look for a polynomial $p_{\mathbf{w}} \in \mathbb{F}[x]$ of a degree at most three, such that $p_{\mathbf{w}}(x_i) = w_i$. We see that for $i \in \{1, 2, 3, 4\}$ it holds $p_{\mathbf{w}}(x_i) = w_i = w_i^{(1)} \cdot w_i^{(2)} \cdot w_i^{(3)} = p_1(x_i) \cdot p_2(x_i) \cdot p_3(x_i) = (p_1 \cdot p_2 \cdot p_3)(x_i)$. The polynomials $p_{\mathbf{w}}$ and $p_1 \cdot p_2 \cdot p_3$ are both of a degree at most three and they are equal in at least four points, hence they are necessarily equal and $\mathrm{Dec}_K(\mathbf{w}) = p_{\mathbf{w}}(y) = (p_1 \cdot p_2 \cdot p_3)(y) = p_1(y) \cdot p_2(y) \cdot p_3(y) = m_1 \cdot m_2 \cdot m_3$ which gives us the multiplicative property for three codewords.

Now let us introduce some notation. We set

$$\mathcal{C}^1 = e_K(\mathcal{P}) = \left\{\mathbf{w} \in \mathbb{F}^4 \mid \exists m \in \mathbb{F} : \Pr[\mathrm{Enc}_K(m) = \mathbf{w}] > 0\right\} \subseteq \mathcal{C} \ .$$

Lemma 9.

$$\mathcal{C}^1 = \left\{\mathbf{w} \in \mathbb{F}^4 \mid \exists p \in \mathbb{F}[x],\ \deg(p) \leq 1, \mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p)\right\} \ ,$$

$\mathcal{C}^1$ is a linear subspace of $\mathbb{F}^4$ isomorphic to the additive group of the ring $(\mathbb{F}^2, +, -, \cdot, (0,0), (1,1))$. (In particular $\mathcal{C}^1$ is a Reed-Solomon code of length n and degree 1.)

Proof. Let $\mathbf{w} \in \mathbb{F}^4$, such that $\mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p_{\mathbf{w}})$, for some $p_{\mathbf{w}} \in \mathbb{F}[x]$, $\deg(p_{\mathbf{w}}) \leq 1$. Then $m := p_{\mathbf{w}}(y) \in \mathbb{F}$ and $\Pr[\mathrm{Ex}_y(m) = p_{\mathbf{w}})] > 0$, hence $\Pr[\mathrm{Enc}_K(m) = \mathbf{w})] > 0$ and $\mathbf{w} \in \mathcal{C}^1$. On the other hand, if $\mathbf{w} = \mathrm{Enc}_K(m)$, then $\mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p_m)$, for some $p_m \in \mathbb{F}[x]$, such that $\deg(p_m) \leq 1$ and $\mathbf{w} \in \{\mathbf{w} \in \mathbb{F}^4 \mid \exists p \in \mathbb{F}[x],\ \deg(p) \leq 1, \mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p)\}$.

We show, that $\mathcal{C}^1$ is a vector space. Let $\mathbf{w}, \mathbf{w}' \in \mathcal{C}^1$. Then $\deg(p_{\mathbf{w}}) \leq 1$ and $\deg(p_{\mathbf{w}'}) \leq 1$. $p_{\mathbf{w}+\mathbf{w}'} = \mathrm{Lag}_{\mathbf{x}}(\mathbf{w} + \mathbf{w}')$ and $\mathrm{Lag}_{\mathbf{x}}$ is an additive homomorphism, so $\mathrm{Lag}_{\mathbf{x}}(\mathbf{w} + \mathbf{w}') = \mathrm{Lag}_{\mathbf{x}}(\mathbf{w}) + \mathrm{Lag}_{\mathbf{x}}(\mathbf{w}')$ and $\deg(\mathrm{Lag}_{\mathbf{x}}(\mathbf{w} + \mathbf{w}')) \leq 1$, therefore $\mathbf{w} + \mathbf{w}' \in \mathcal{C}^1$. Similarly, we can show that $a \cdot \mathbf{w} \in \mathcal{C}^1$ for all $a \in \mathbb{F}$ and for all $\mathbf{w} \in \mathcal{C}^1$. The fact, that $\mathcal{C}^1$ is a Reed-Solomon code of length four and degree one comes straight from the definition of Reed-Solomon codes.

Now we show, that the mapping

$$\begin{array}{rccc} \varphi: & \mathcal{C}^1 & \longrightarrow & \mathbb{F}^2 \\ & (w_1, w_2, w_3, w_4) & \longmapsto & (w_1, w_2) \end{array}$$

is an isomorphism of additive groups. It is obvious, that φ is a homomorphism. We have $\mathrm{Ker}(\varphi) = \{\mathbf{w} \in \mathcal{C}^1 \mid w_1 = w_2 = 0\}$. Let $\mathbf{w} \in \mathrm{Ker}(\varphi)$ and $p_{\mathbf{w}} = a \cdot x + b$ for some $a, b \in \mathbb{F}$.

$$\begin{aligned} w_1 =& p_{\mathbf{w}}(x_1) = 0 = ax_1 + b \\ w_2 =& p_{\mathbf{w}}(x_2) = 0 = ax_2 + b \\ & \Rightarrow\ ax_1 = ax_2 \text{ and as } x_1 \neq x_2\ \Rightarrow a, b = 0 \ . \end{aligned}$$

Hence $\mathbf{w} = (0,0,0,0)$, $\mathrm{Ker}(\varphi) = \{(0,0,0,0)\}$ and φ is injective. Now let $(w_1, w_2) \in \mathbb{F}^2$. Then set

$$p := \frac{w_1 - w_2}{x_1 - x_2} x + \frac{w_2 x_1 - w_1 x_2}{x_1 - x_2} \ .$$

We have $\deg(p) \leq 1$, it is well defined since $x_1 \neq x_2$. We set $\widetilde{\mathbf{w}} := \mathrm{Ev}_{\mathbf{x}}(p)$. Then $\widetilde{\mathbf{w}} \in \mathcal{C}^1$, $\widetilde{w}_1 = w_1$ and $\widetilde{w}_2 = w_2$. We conclude, that φ is surjective. □

Similarly, we define $\mathcal{C}^2$ and $\mathcal{C}^3$,

$$\mathcal{C}^2 = \left\{\mathbf{w} \in \mathbb{F}^4 \mid \exists p \in \mathbb{F}[x],\ \deg(p) \leq 2, \mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p)\right\}$$
$$\mathcal{C}^3 = \left\{\mathbf{w} \in \mathbb{F}^4 \mid \exists p \in \mathbb{F}[x],\ \deg(p) \leq 3, \mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p)\right\}$$

Note that $\mathcal{C}^2$ and $\mathcal{C}^3$ are linear codes, $\mathcal{C}^2 \cong \mathbb{F}^3$ and $\mathcal{C}^3 = \mathbb{F}^4$. We have the following chain of subcodes:

$$\mathcal{C}^1 \subsetneq \mathcal{C}^2 \subsetneq \mathcal{C}^3 = \mathbb{F}^4$$

The fact that $\mathcal{C}^i \neq \mathcal{C}^{i+1}$ is to be shown in Proposition 14. The elements of the linear code $\mathcal{C}^1$ are exactly the evaluation vectors of all polynomials of degree at most one, the elements of $\mathcal{C}^2$ are the evaluations of polynomials of degree at most two. If we multiply at most two polynomials of degree at most one, we get a polynomial of a degree at most two. Analogically, if we multiply at most two codewords from $\mathcal{C}^1$, we get a codeword from $\mathcal{C}^2$, which can still be decoded back to the product of the original messages.

Similarly, the product of up to three codewords from $\mathcal{C}^1$ is a codeword in $\mathcal{C}^3$ and may be decoded correctly. Also the product of a codeword from $\mathcal{C}^1$ and a codeword from $\mathcal{C}^2$ is a codeword in $\mathcal{C}^3$ and may be well decoded.

In the following counter-example, we show, that if we take more than three codewords from $\mathcal{C}^1$, it may happen, that the decoding of their product does not equal the product of their decoding.

Example 10 (Counter-example)**.** *Fix $K = (\mathbf{x}, y) = ((0,1,2,3), 4)$. We take four messages in $\mathbb{F}_5$ and we encode them to get four codewords from $\mathcal{C}^1$.*

$$\begin{array}{lll} m_1 = 1 & p_1 = x + 2 & \mathbf{w}^{(1)} = (2,3,4,0) \\ m_2 = 3 & p_2 = x + 4 & \mathbf{w}^{(2)} = (4,0,1,2) \\ m_3 = 3 & p_3 = 2x & \mathbf{w}^{(3)} = (0,2,4,1) \\ m_4 = 2 & p_4 = 3x & \mathbf{w}^{(4)} = (0,3,1,4) \end{array}$$

We have $\prod_{j=1}^{4} \left(\mathrm{Dec}_K(\mathbf{w}^{(j)})\right) = \prod_{j=1}^{4} m_j = 1 \cdot 3 \cdot 3 \cdot 2 = 3$. We set

$$\mathbf{w} := \prod_{j=1}^{4} \mathbf{w}^{(j)} = (0,0,1,0)$$

$$\mathrm{Dec}_K\left(\prod_{j=1}^{4} \mathbf{w}^{(j)}\right) = \mathrm{Dec}_K(\mathbf{w}) = \mathrm{Ev}_y\left(\mathrm{Lag}_{\mathbf{x}}(0,0,1,0)\right)$$
$$= \mathrm{Ev}_y(2x^3 + 2x^2 + x) = 2y^3 + 2y^2 + y = 4$$

We get

$$3 = \prod_{j=1}^{4} \left(\mathrm{Dec}_K(\mathbf{w}^{(j)})\right) \neq \mathrm{Dec}_K\left(\prod_{j=1}^{4} \mathbf{w}^{(j)}\right) = 4$$

The reason why these two expressions do not equal is the fact that $\mathrm{Lag}_{\mathbf{x}}(\mathbf{w}) \neq \prod_{j=1}^{4} \mathrm{Lag}_{\mathbf{x}}(\mathbf{w}^{(j)})$. *More precisely,* $\mathrm{Lag}_{\mathbf{x}}(\mathbf{w}) = \prod_{j=1}^{4} \mathrm{Lag}_{\mathbf{x}}(\mathbf{w}^{(j)}) \mod \prod_{j=1}^{4}(x - x_i)$, *i.e.*

$$\mathrm{Lag}_{\mathbf{x}}(\mathbf{w}) = \prod_{j=1}^{4} \mathrm{Lag}_{\mathbf{x}}(\mathbf{w}^{(j)}) + q(x) \cdot \prod_{i=1}^{4}(x - x_i) \ ,$$

for some $q(x) \in \mathbb{F}[x]$. *Hence*

$$\begin{aligned}\mathrm{Dec}_K(\mathbf{w}) = \mathrm{Lag}_{\mathbf{x}}(\mathbf{w})(y) &= \prod_{j=1}^{4} \mathrm{Lag}_{\mathbf{x}}(\mathbf{w}^{(j)})(y) + q(y) \cdot \prod_{i=1}^{4}(y - x_i) \\ &= \prod_{j=1}^{4} \mathrm{Dec}_K\, \mathbf{w}^{(j)} + q(y) \cdot a \ ,\end{aligned}$$

for some $a \in \mathbb{F}^*$. a *is non-zero, because* y *is distinct from all* x_i*s. We see, that* $\mathrm{Dec}_K(\mathbf{w})$ *does not equal* $\prod_{j=1}^{4} \mathrm{Dec}_K\, \mathbf{w}^{(j)}$, *whenever* $q(y)$ *is non-zero. In this example,* $q(x) = 1$ *and* $q(y) = 1$.

Clearly, $\mathcal{C}^4 = \{\mathbf{w} \in \mathbb{F}^4 \mid \exists p \in \mathbb{F}[x],\ \deg(p) \leq 4, \mathbf{w} = \mathrm{Ev}_{\mathbf{x}}(p)\}$ is a subset of $\mathbb{F}^4$, but we can no longer ensure, that all codewords in $\mathcal{C}^4$ are to be decoded correctly.

We have seen, that the scheme is 3-multiplicative. We return to the suggested problem: We want to outsource a computation of $m_1 \cdot m_2 \cdot m_3$, without providing any information about the data. We make use of the cryptosystem the following way: We send the company vectors $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)} = \mathrm{Enc}_K(m_1)$, $\mathrm{Enc}_K(m_2)$, $\mathrm{Enc}_K(m_3)$. They calculate the product $\mathbf{w} = \prod_{j=1}^{3} \mathrm{Enc}_K(m_j)$ and send it back to us. We compute $\mathrm{Dec}_K(\mathbf{w})$ and we get the desired product $m_1 \cdot m_2 \cdot m_3$, without ever revealing $m_1, m_2, m_3, m_1 \cdot m_2 \cdot m_3$.

If we need to calculate the product of one codeword from $\mathcal{C}^1$ and another from $\mathcal{C}^2$, it is still fine, but imagine, we ask the company for a product of $\mathbf{w}^{(1)}, \mathbf{w}^{(2)} \in \mathcal{C}^2$. We would likely prefer them to return an error, as $\mathbf{w}^{(1)} \cdot \mathbf{w}^{(2)}$ is not necessarily in $\mathcal{C}^3$ and we cannot be sure of the correctness of its decryption.

Unless we provide the company with some extra information, they cannot tell the codewords from $\mathcal{C}^1, \mathcal{C}^2$ and $\mathcal{C}^3$ apart, as they cannot calculate the underlying polynomial $p_{\mathbf{w}}$ and see its degree. However, we do not want to provide them with information about the secret key. Instead we can send pairs in the form $(\mathbf{w}, \deg(p_{\mathbf{w}}))$. The company then returns $\begin{cases} \mathbf{w}' \cdot \mathbf{w}' & \text{if } \deg(p_{\mathbf{w}'}) + \deg(p_{\mathbf{w}'}) \leq 3, \\ error & \text{if } \deg(p_{\mathbf{w}'}) + \deg(p_{\mathbf{w}'}) > 3 \ . \end{cases}$

There is an obvious limitation of this scheme. We can only multiply at most three encodings to keep the correctness. How do we overcome this limitation? Well, we take a larger field $\mathbb{F}$ and a longer vector $\mathbf{x}$, say $length(\mathbf{x}) = n$ and then

we can do up to $n-1$ multiplications. No matter what n we take, we are still limited and the cost is high in terms of computation complexity. The length of codewords is linear with the number of possible multiplications.

2.2 Scheme proposed by Armknecht et al.

In July 2011 in [AAPS11] Frederik Armknecht, Daniel Augot, Ludovic Perret and Ahmad-Reza Sadeghi proposed a code-based scheme for symmetric encryption that is homomorphic with respect to addition and in a limited way it is also multiplicatively homomorphic.

The paper describes a generic construction based on so called *special evaluation codes* and presents an instantiation of these by Reed-Muller codes. We shall describe the instantiation by Reed-Solomon codes directly. The scheme uses the natural property of linear codes being additive groups. It achieves a limited multiplicativity, implementing a scheme, that is similar to the one we described in Section 2.1.

Algorithm 1 describes the cryptosystem proposed by Armknecht et al., instantiated by the Reed-Solomon codes. The set of messages $\mathcal{P} = \mathbb{F}$, the set of ciphertexts $\mathcal{C} = \mathbb{F}^n$ and the keys $K \in \mathcal{K}$ are triples $(\mathbf{x}, y, I)$, where $\mathbf{x} \in \mathbb{F}^n$, $y \in \mathbb{F}$ and $I \subseteq \{1, \ldots, n\}$.

The vector $\mathbf{x} \in \mathbb{F}^n$ is called codeword support, $y \in \mathbb{F}$ is called message support and the set I is called the set of good locations. We shall denote $\mathrm{Enc}_K(m) = \mathrm{Encrypt}(m, K)$ and $\mathrm{Dec}_K(c) = \mathrm{Decrypt}(c, K)$.

Proposition 11. *Algorithm 1 describes a cryptosystem.*

Proof. Let us check the four conditions from the definition of a cryptosystem.

1. $\mathcal{P} = \mathbb{F}$ is a finite set.
2. $\mathcal{C} \subset \mathbb{F}^n$ is a finite set.
3. The key-space

   $$\mathcal{K} = \{(\mathbf{x}, y, I) \mid \mathbf{x} \in \mathbb{F}^n, y \in \mathbb{F}, I \subset \{1, ..., n\}, x_i \neq x_j \neq y, \forall i, j \leq n, |I| = k)\}$$

   is a finite set of possible keys. Note that it is non-empty as we can always choose $n+1$ distinct elements of $\mathbb{F}_q$ since $n < q$ and also $k \leq n$.
4. Let $m \in \mathbb{F}$, $K = (\mathbf{x}, y, I) \in \mathcal{K}$, $c = \mathrm{Enc}_K(m)$. First note that the condition $e_i = 0$ for all $i \in I$ implies that $\mathrm{Dec}_K(c) = \mathrm{Dec}_K(\mathbf{w} + e) = \mathrm{Dec}_K(\mathbf{w})$, where $\mathbf{w} = \mathrm{Encode}(m, \mathbf{x}, y)$, as the polynomial p_c only depends on the evaluations at the good locations.

   $$\mathbf{w} = (w_1, ..., w_n) = (p(x_1), ..., p(x_n)) \,,\, p_c(x_i) = w_i = p(x_i),\ \forall i \in I$$

   The polynomials p and p_c evaluate equally in at least k points. They are both of a degree less than k as $\deg(p) < d = \left\lfloor \frac{k}{\mu} \right\rfloor \leq k$. Necessarily the two

Algorithm 1 Code-based Homomorphic Encryption Scheme

SETUP

Input: (n, k, q, μ), such that $n, k, q, \mu \in \mathbb{N}$, q is a prime power, $k \leq n$, $n < q$, $2k \geq \mu$

Output: $K = (\mathbf{x}, y, I)$, $\mathbf{x} \in \mathbb{F}^n, y \in \mathbb{F}, I \subset \{1, \ldots, n\}$, $|I| = k$

$\mathbb{F} := \mathbb{F}_q$
for $i = 1 \to n$ **do**
choose $x_i \leftarrow \mathbb{F} \setminus \{x_1, \ldots, x_{i-1}\}$
end for
set $\mathbf{x} := (x_1, \ldots, x_n) \in \mathbb{F}^n$
choose $y \leftarrow \mathbb{F} \setminus \{x_1, ..., x_n\}$
choose a random $I \subset \{1, ..., n\}$, such that $|I| = k$
return the key $K = (\mathbf{x}, y, I)$

ENCODE

Input: $m \in \mathbb{F}$, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$, $y \in \mathbb{F}$

Output: $\mathbf{w} \in \mathbb{F}^n$

choose $p \leftarrow \mathbb{F}[x]$ such that $\deg(p) < d = \left\lfloor \frac{k}{\mu} \right\rfloor$
set $p := p - p(y) + m$
for $i = 1 \to n$ **do**
$w_i := p(x_i)$
end for
set $\mathbf{w} := (w_1, ..., w_n) \in \mathbb{F}^n$
return w

ENCRYPT

Input: $m \in \mathbb{F}, K = (\mathbf{x}, y, I)$

Output: $c \in \mathbb{F}^n$

for $i = 1 \to n$ **do**
if $i \in I$ **then**
set $e_i := 0 \in \mathbb{F}$
else
choose $e_i \leftarrow \mathbb{F}$
end if
end for
set the ciphertext $c = (c_1, ..., c_n) := Encode(m, \mathbf{x}, y) + (e_1, \ldots, e_n) \in \mathbb{F}^n$
return c

DECRYPT

Input: $c \in \mathbb{F}^n, K = (\mathbf{x}, y, I)$

Output: $m \in \mathbb{F}$

find the polynomial $p_c \in \mathbb{F}[x]$, such that $\deg(p_c) < k$ and $p_c(x_i) = c_i$ for all $i \in I$
set $m := p_c(y)$
return m

polynomials are equal. In particular, $p_c(y) = p(y)$. The polynomial p has

been chosen in such way that $p(y) = m$. We can recapitulate:

$$\begin{aligned}\mathrm{Dec}_K\left(\mathrm{Enc}_K(m)\right) &= \mathrm{Dec}_K(c) = \mathrm{Dec}_K(\mathbf{w}+e)\\ &= \mathrm{Dec}_K(\mathbf{w}) = p_c(y) = p(y) = m.\end{aligned}$$

As the key K and the message m were chosen arbitrarily, the Condition 4 of Definition 1 is satisfied, which concludes the proof. □

Proposition 12. *The cryptosystem, described by Algorithm 1 is additively homomorphic and μ-multiplicative.*

Proof. Clearly, the cryptosystem is additively homomorphic. Let $l \in \mathbb{N}$, $l < \mu$ and $c^{(1)}, ..., c^{(l)} \in \mathrm{Enc}_K(\mathbb{F})$. We need to show that

$$\mathrm{Dec}_K\left(\prod_{j=1}^{l} c^{(j)}\right) = \prod_{j=1}^{l} \mathrm{Dec}_K\left(c^{(j)}\right)$$

Let $c = (c_1, ..., c_n) := \prod_{j=1}^{l} c^{(j)}$. Hence

$$\mathrm{Dec}_K\left(\prod_{j=1}^{l} c^{(j)}\right) = \mathrm{Dec}_K(c) = p_c(y) \ ,$$

where p_c is such polynomial, that for all $i \in I$ it holds $p_c(x_i) = c_i$ and $\deg(p_c) < k$. On the other hand,

$$\prod_{j=1}^{l} \mathrm{Dec}_K\left(c^{(j)}\right) = \prod_{j=1}^{l} p_{c^{(j)}}(y) = \left(\prod_{j=1}^{l} p_{c^{(j)}}\right)(y) \ ,$$

where $\left(\prod_{j=1}^{l} p_{c^{(j)}}\right)$ is such polynomial that for all $i \in I$ it holds $\left(\prod_{j=1}^{l} p_{c^{(j)}}\right)(x_i) = \prod_{j=1}^{l} p_{c^{(j)}}(x_i) = \prod_{j=1}^{l} c_i^{(j)} = c_i$ and

$$\deg\left(\prod_{j=1}^{l} p_{c^{(j)}}\right) = \sum_{j=1}^{l} \deg\left(p_{c^{(j)}}\right) \leq l \cdot (d-1) < k \ .$$

We see that the two polynomials p_c and $\left(\prod_{j=1}^{l} p_{c^{(j)}}\right)$ equal in at least k points and both are of a degree less than k, therefore they are equal and $p_c(y) = \left(\prod_{j=1}^{l} p_{c^{(j)}}\right)(y)$. We have

$$\begin{aligned}\mathrm{Dec}_K\left(\prod_{j=1}^{l} c^{(j)}\right) &= \mathrm{Dec}_K(c) = p_c(y) = \left(\prod_{j=1}^{l} p_{c^{(j)}}\right)(y)\\ &= \prod_{j=1}^{l} p_{c^{(j)}}(y) = \prod_{j=1}^{l} \mathrm{Dec}_K\left(c^{(j)}\right) \ .\end{aligned}$$

□

Let $K = (\mathbf{x}, y, I) \in \mathcal{K}$. We denote $\widetilde{\mathbf{x}} = (\widetilde{x}_1, ..., \widetilde{x}_k) := (x_{i_1}, \ldots, x_{i_k})$, where $\{i_1, \ldots, i_k\} = I$. We denote $\mathcal{C}^1 \subset \mathbb{F}_q^k$ a punctured Reed-Solomon code $RS_q(d)^*$ with the codeword support $\widetilde{\mathbf{x}} = (\widetilde{x}_1, ..., \widetilde{x}_k)$, i.e.

$$\mathcal{C}^1 = \{(f(\widetilde{x}_1), ..., f(\widetilde{x}_k)) \mid f \in \mathbb{F}_q[x], \deg(f) < d\}$$

For $l \in \mathbb{N}$ we denote $\mathcal{C}^l = RS_q\,(d \cdot l)^*$. Again we use the same codeword support $\widetilde{\mathbf{x}}$.

Lemma 13. *Let* $\mathrm{Enc}_K(\mathbb{F}) = \{c \in \mathcal{C} \mid \exists m \in \mathbb{F} : \Pr[\mathrm{Enc}_K(m) = c] > 0\}$. *Then*

$$\mathcal{C}^1 = \{\widetilde{\mathbf{w}} = (\widetilde{w}_1, \ldots, \widetilde{w}_k) \in \mathbb{F}^k \mid \exists \mathbf{w} \in \mathrm{Enc}_K(\mathbb{F}) : w_{i_j} = \widetilde{w}_j,\ \forall j \in I\}\ .$$

In other words, $\mathcal{C}^1$ *is the set of possible fresh encryptions truncated at the good locations.*

These are exactly such vectors in $\mathbb{F}^n$ that if we interpolate them in the good locations of $\mathbf{x}$, we get a polynomial that has a degree at most $d-1$. The proof of this lemma is analogical to the proof of Lemma 9.

Proposition 14. *Let* $\mathcal{C}^1, ..., \mathcal{C}^\mu$ *be defined as above. Then*

$$\mathcal{C}^1 \subsetneq \mathcal{C}^2 \subsetneq ... \subsetneq \mathcal{C}^\mu$$

Proof. The fact, that $\mathcal{C}^1 \subseteq \mathcal{C}^2 \subseteq ... \subseteq \mathcal{C}^\mu$ is trivial. Let $1 < l \leq \mu$. Let $p \in \mathbb{F}[x]$, such that $\deg(p) = d \cdot l - 1$. Then $\mathbf{w} = \mathrm{Ev}_{\widetilde{\mathbf{x}}}(p) \in \mathcal{C}^l$. Suppose, that $\mathbf{w} \in \mathcal{C}^{l-1}$. Then there is a polynomial $q \in \mathbb{F}[x]$, $\deg(q) \leq d \cdot (l-1) - 1$. Such q evaluates equally as p in at least k points. As $k > \deg(q) + 1$, we get that $p = q$, which is in contradiction with $\deg(p) > \deg(q)$. Therefore $\mathbf{w} \notin \mathcal{C}^{l-1}$ and

$$\mathcal{C}^1 \neq \mathcal{C}^2 \neq ... \neq \mathcal{C}^\mu.$$

□

As we noted in the proof of Proposition 11, the corrupt locations of a ciphertext do not affect the decryption. If $c = (c_1, \ldots, c_n)$ and we denote $\tilde{c} = (c_{j_1}, \ldots, c_{j_k}), \{j_1, \ldots, j_k\} = I$, then the function Decrypt may be expressed as follows:

$$\mathrm{Dec}_{(\mathbf{x},y,I)}(c) = \mathrm{Dec}_{(\tilde{\mathbf{x}},y,\{1,\ldots,k\})}(\tilde{c})\ .$$

We see, that in the context of a fixed I we can naturally identify c with $\tilde{c}$, $\mathbf{w}$ with $\widetilde{\mathbf{w}}$ and n with k. We shall do so in the following section.

The idea of corrupt positions has been used so that the cryptosystem is more difficult to break. In [AAPS11] Armknecht et. al. present a reduction of the problem of breaking this cryptosystem to the problem of decoding in a random code. We do not describe the reduction in the thesis.

2.3 Overcoming μ-limitation

We would like to modify the scheme in such way that we could perform an unlimited number of multiplications on codewords of fixed length, say $length(\mathbf{w}) = n$. Let us have a look at the scheme and think what we could do about the functions Expression and Evaluation.

$$\mathbb{F} \underset{\mathrm{Ex}^{-1}}{\overset{\mathrm{Ex}}{\rightleftarrows}} \mathbb{F}[x] \underset{\mathrm{Lag}}{\overset{\mathrm{Ev}}{\rightleftarrows}} \mathbb{F}^n$$

As we explained in Section 1.2, the composition of $\mathrm{Lag}_{\mathbf{x}}$ and $\mathrm{Ev}_{\mathbf{x}}$ is an identity on $\mathcal{L} = \mathbb{F}[x]/(\prod_{i=1}^{n}(x - x_i))$ and the function $\mathrm{Ev}_{\mathbf{x}}$ is a ring isomorphism of $\mathcal{L}$ and $\mathbb{F}^n$. This seems fine, because it supports the two operations $(+, \cdot)$, just like we intended. Now we have a look at the function Ex and whether there is a way, we could modify it to overcome the limitation of μ-multiplicativity. We need to find a ring homomorphism $\phi : \mathcal{L} \longrightarrow \mathbb{F}$, so that $\mathrm{Dec}_K = \phi \circ \mathrm{Lag}_{\mathbf{x}}$ is a homomorphism of rings. We will show, that such homomorphism ϕ needs to maintain the equivalence on $\mathbb{F}[x]$ induced by the function $\mathrm{Ev}_{\mathbf{x}}$, i.e. if $\prod_{i=1}^{n}(x - x_i) \mid (f_1 - f_2)$, then $\phi(f_1) = \phi(f_2)$. Then we shall explore other properties of the homomorphism Dec_K, under the requirement of additivity and full multiplicativity. Let us assume that there is a ring homomorphism ϕ. We have

$$\begin{aligned}
\mathrm{Dec}_K : \mathbb{F}^n \xrightarrow{\mathrm{Lag}_{\mathbf{x}}} \mathbb{F}[x] \xrightarrow{\phi} \mathbb{F} \quad &\text{and} \\
\mathrm{Dec}_K\left(c^{(1)} \cdot c^{(2)}\right) &= \phi\left(\mathrm{Lag}_{\mathbf{x}}\left(c^{(1)} \cdot c^{(2)}\right)\right) \\
&= \phi\left(\mathrm{Lag}_{\mathbf{x}}\left(c^{(1)}\right) \cdot \mathrm{Lag}_{\mathbf{x}}\left(c^{(2)}\right) \mod \prod_{i=1}^{n}(x - x_i)\right) \\
&= \phi\left(\mathrm{Lag}_{\mathbf{x}}\left(c^{(1)}\right) \cdot \mathrm{Lag}_{\mathbf{x}}\left(c^{(2)}\right) - q(x) \cdot \prod_{i=1}^{n}(x - x_i)\right)
\end{aligned}$$

Since ϕ is a ring homomorphism, we get

$$\begin{aligned}
\mathrm{Dec}_K\left(c^{(1)} \cdot c^{(2)}\right) &= \phi\left(\mathrm{Lag}_{\mathbf{x}}\left(c^{(1)}\right)\right) \cdot \phi\left(\mathrm{Lag}_{\mathbf{x}}\left(c^{(2)}\right)\right) - \phi(q(x)) \cdot \phi\left(\prod_{i=1}^{n}(x - x_i)\right) \\
&= \mathrm{Dec}_K\left(c^{(1)}\right) \cdot \mathrm{Dec}_K\left(c^{(2)}\right) - \phi(q(x)) \cdot \phi\left(\prod_{i=1}^{n}(x - x_i)\right) \quad .
\end{aligned}$$

We see that ϕ needs to satisfy $\phi(\prod_{i=1}^{n}(x - x_i)) = 0$, i.e. $\prod_{i=1}^{n}(x - x_i) \in \mathrm{Ker}(\phi)$. It is a necessary condition for Dec_K to be a homomorphism of rings.

Let us have a look at other conditions Dec_K needs to satisfy. We have

$$\mathrm{Dec}_K : \mathbb{F}^n \longrightarrow \mathbb{F} \quad .$$

It is a linear mapping, so it may be described by its values on the basis of $\mathbb{F}^n$. Let $e_i \in \mathbb{F}^n$ be a vector of zeros at all positions but the i-th position, where it is equal to 1. Then $\{e_1, \ldots, e_n\}$ is a basis of $\mathbb{F}^n$ and we denote $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n$ the vector, such that

$$\mathrm{Dec}_K(e_i) = a_i, \ i = 1, \ldots, n$$

Dec_K is a multiplicative homomorphism, so

$$\mathrm{Dec}_K(e_i \cdot e_j) = \mathrm{Dec}_K(e_i) \cdot \mathrm{Dec}_K(e_j) \ , \ \forall i, j \in \{1, \ldots, n\}$$

when $i \neq j$, we get

$$\begin{aligned} &\mathrm{Dec}_K(e_i \cdot e_j) = \mathrm{Dec}_K(0) = 0 \\ &\mathrm{Dec}_K(e_i) \cdot \mathrm{Dec}_K(e_j) = a_i \cdot a_j \ , \text{ so} \\ &a_i \cdot a_j = 0 \ \forall i \neq j \ . \end{aligned}$$

We see, that there is at most one $a_\alpha \in \{a_1, \ldots, a_n\}$, such that $a_\alpha \neq 0$. we have

$$\begin{aligned} &\mathrm{Dec}_K(e_\alpha \cdot e_\alpha) = \mathrm{Dec}_K(e_\alpha) = a_\alpha \\ &\mathrm{Dec}_K(e_\alpha) \cdot \mathrm{Dec}_K(e_\alpha) = a_\alpha \cdot a_\alpha \ , \text{ so} \\ &a_\alpha^2 = a_\alpha \quad \text{and } a_\alpha = 1 \ . \end{aligned}$$

We get that either $\mathrm{Dec}_K \equiv 0$, which is not very convenient, or $\mathrm{Dec}_K(\mathbf{w})$ only depends on the α-th position of $\mathbf{w} = (w - 1, \ldots, w_\alpha, \ldots, w_n)$ and

$$\mathrm{Dec}_K(\mathbf{w}) = \mathrm{Dec}_K(0, \ldots, 0, w_\alpha, 0, \ldots, 0) = w_\alpha \cdot \mathrm{Dec}_K(e_\alpha) = w_\alpha \ .$$

This means that the plaintext is an open part of the ciphertext, and it is always placed at the same position - such cryptosystem would be easily broken.

When we look back at the homomorphism $\phi : \mathbb{F}[x] \longrightarrow \mathbb{F}$,

$$\begin{array}{rccc} \phi : & \mathbb{F}[x] & \longrightarrow & \mathbb{F} \\ & p & \longmapsto & m \ , \end{array}$$

We see that $\phi = \mathrm{Dec}_K \circ \mathrm{Ev}_{\mathbf{x}}$ and

$$\phi(p) = \mathrm{Dec}_K(\mathrm{Ev}_{\mathbf{x}}(p)) = \mathrm{Dec}_K(p(x_1), \ldots, p(x_n)) = p(x_\alpha) \ ,$$

so ϕ is the homomorphism that evaluates polynomials in the point x_α, where $x_\alpha \in \{x_1, \ldots, x_n\} \in \mathbb{F}$.
Note, that $\prod_{i=1}^{n}(x - x_i)$ is indeed in $\mathrm{Ker}(\phi)$.

We conclude, that the limitation on the number of possible multiplications cannot be overcome by modification of the Expression function.

2.4 Linearly algebraic perspective

In this section we take a further look at the function Dec_K and propose a very simple attack on the cryptosystem. The possibility of this attack has been identified by Armknecht et. al. in [AAPS11].

Proposition 15. *For a key $K \in \mathcal{K}$, $K = (\mathbf{x}, y, I)$ there exists a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n$, such that*

$$\mathrm{Dec}_K(\mathbf{w}) = \sum_{i=1}^{n} w_i \cdot a_i \ , \ \textit{for all } \mathbf{w} \in \mathbb{F}^n \ .$$

Proof. Set

$$a_i = \begin{cases} 0 & \text{if } i \notin I \\ \prod_{j \in I,\ j \neq i} \left(\frac{y - x_i}{x_j - x_i} \right) & \text{if } i \in I \ . \end{cases}$$

Then

$$\begin{aligned} \sum_{i=1}^{n} w_i \cdot a_i = \sum_{i \in I} w_i \cdot a_i = \sum_{i \in I} w_i \prod_{j \in I,\ j \neq i} \left(\frac{y - x_i}{x_j - x_i} \right) \\ = \left(\sum_{i \in I} w_i \prod_{j \in I,\ j \neq i} \left(\frac{x - x_i}{x_j - x_i} \right) \right) (y) \ . \end{aligned}$$

We denote $\widetilde{\mathbf{w}} = (w_i)_{i \in I}$, $\widetilde{\mathbf{x}} = (x_i)_{i \in I}$. Then

$$\sum_{i=1}^{n} w_i \cdot a_i = \mathrm{Lag}_{\widetilde{\mathbf{x}}}(\widetilde{\mathbf{w}})(y) = \mathrm{Dec}_K(\mathbf{w}) \ .$$

□

The attack works as follows. Attacker collects n pairs of plaintext-ciphertext: $\{(m_1, c_1), \ldots, (m_n, c_n)\}$. It holds $\sum_{i=1}^{n} c_{ij} a_i = m_j$, for $j = 1, \ldots, n$. We have n linear equations in n variables. If the ciphertexts are linearly independent then the system of equations has a unique solution $\mathbf{a}$. Then any ciphertext $c \in \mathbb{F}^n$ may be decrypted as $\mathrm{Dec}_K(c) = \sum_{i=1}^{n} c_i a_i$.

We see that the cryptosystem only allows us to safely encrypt at most $n - 1$ messages with one key. This is a very strong limitation.

We note, that this type of an attack may be applied to any additively homomorphic encryption scheme, where $\mathcal{C}$ has a finite dimension. In this cryptosystem, the attack is especially feasible as the dimension of the ciphertext space is particularly low.

3. Gröbner bases introduction

3.1 Orderings

The following definitions appear in [Win96]. We reformulate them to meet the notation of this thesis.

Definition 16. *Let $[X]$ be the set of terms in n variables, i.e.*

$$[X] = \left\{ x_1^{i_1} \cdots x_n^{i_n} \mid i_j \in \mathbb{N}_0, j = 1, \ldots, n \right\}.$$

An ordering $<$ on $[X]$ is called admissible if it satisfies the following two conditions:

1. $1 = x_1^0 \cdots x_n^0 < t$, *for all* $t \in [X] \setminus \{1\}$
2. $s < t \Rightarrow su < tu$, *for all* $s, t, u \in [X]$.

Definition 17. *Let π be a permutation on $\{1, \ldots, n\}$. We define the lexicographic ordering with $x_{\pi(1)} > \ldots > x_{\pi(n)}$ in the following way:*

$$x_1^{i_1} \cdots x_n^{i_n} <_{lex,\pi} x_1^{j_1} \cdots x_n^{j_n}, \text{ if there exists such } l \in \{1, \ldots, n\}, \text{ that}$$

$$i_{\pi(k)} = j_{\pi(k)}, \text{ for all } k \in \mathbb{N}, k < l \text{ and } i_{\pi(l)} < j_{\pi(l)}.$$

Definition 18. *The graduated lexicographic ordering with respect to permutation π and a weight function $w : \{1, \ldots, n\} \longrightarrow \mathbb{R}^+$ is defined as follows:*

$$x_1^{i_1} \cdots x_n^{i_n} <_{glex,\pi,w} x_1^{j_1} \cdots x_n^{j_n}, \text{ if one of the following is satisfied:}$$

1. $\sum_{k=1}^{n} w(k) i_k < \sum_{k=1}^{n} w(k) j_k$
2. $\sum_{k=1}^{n} w(k) i_k = \sum_{k=1}^{n} w(k) j_k$ *and* $x_1^{i_1} \cdots x_n^{i_n} <_{lex,\pi} x_1^{j_1} \cdots x_n^{j_n}$.

Proposition 19. *The lexicographic ordering with respect to π and the graduated lexicographic ordering with respect to π and w are admissible orderings.*

Proof. 1_{lex} Let $s \in [X] \setminus \{1\}$. We have $s = x_1^{i_1} \cdots x_n^{i_n}$ and there exists some $k \in \{1, \ldots, n\}$, such that $i_k > 0$. We set $l \in \{1, \ldots, n\}$ the smallest number that satisfies $i_{\pi(l)} > 0$. Then l satisfies $0 = i_{\pi(k)}$ for all $k < l$ and $0 < i_{\pi(l)}$, i.e. $1 <_{lex,\pi} s$.

2_{lex} Let $s, t, u \in [X]$. We can write these as $s = x_1^{i_1} \cdots x_n^{i_n}$, $t = x_1^{j_1} \cdots x_n^{j_n}$ and $u = x_1^{m_1} \cdots x_n^{m_n}$. Then $su = x_1^{i_1+m_1} \cdots x_n^{i_n+m_n}$ and $tu = x_1^{j_1+m_1} \cdots x_n^{j_n+m_n}$. Since $s <_{lex,\pi} t$, there exists such l that $i_{\pi(k)} = j_{\pi(k)}$, for all $k \in \mathbb{N}, k < l$ and $i_{\pi(l)} < j_{\pi(l)}$. This implies that l also satisfies $i_{\pi(k)} + m_{\pi(k)} = j_{\pi(k)} + m_{\pi(k)}$, for all $k \in \mathbb{N}, k < l$ and $i_{\pi(l)} + m_{\pi(l)} < j_{\pi(l)} + m_{\pi(l)}$, i.e. $su <_{lex,\pi} tu$.

We have shown that $<_{lex,\pi}$ is an admissible ordering on $[X]$.

1_{glex} Let $s \in [X] \setminus \{1\}$. Then $s = x_1^{i_1} \cdots x_n^{i_n}$, where $i_j \in \mathbb{N}_0$ and there exists an $l \in \{1, \ldots, n\}$, such that $i_l > 0$. We have $w(k) > 0, i_k \geq 0$ for all $k = 1, \ldots, n$, so

$$\sum_{k=1}^{n} w(k)i_k = \sum_{k=1, k\neq l}^{n} w(k)i_k + w(l)i_l \geq w(l)i_l > 0 = \sum_{k=1}^{n} w(k) \cdot 0$$

and $1 < s$.

2_{glex} Let s, t, u be defined as in the lexicographic case. $s <_{glex,\pi,w} t$, so either $\sum_{k=1}^{n} w(k)i_k < \sum_{k=1}^{n} w(k)j_k$ or $\sum_{k=1}^{n} w(k)i_k = \sum_{k=1}^{n} w(k)j_k$ and $s <_{lex,\pi} t$. In the first case we have

$$\begin{aligned}\sum_{k=1}^{n} w(k)(i_k + m_k) &= \sum_{k=1}^{n} w(k)i_k + \sum_{k=1}^{n} w(k)m_k \\ &< \sum_{k=1}^{n} w(k)j_k + \sum_{k=1}^{n} w(k)m_k = \sum_{k=1}^{n} w(k)(j_k + m_k) \ .\end{aligned}$$

In the other case we have $\sum_{k=1}^{n} w(k)(i_k + m_k) = \sum_{k=1}^{n} w(k)(j_k + m_k)$ and $su <_{lex,\pi} tu$, because $<_{lex,\pi}$ is an admissible ordering as shown earlier in the proof. We conclude, that $su <_{glex,\pi,w} tu$ and that $<_{glex,\pi,w}$ is an admissible ordering.

□

Notation 20. *Let K be a field. Fix an admissible ordering $<$ on $[X]$. Let $f \in K[x_1, \ldots, x_n]$. We denote $\mathrm{lt}(f), \mathrm{lc}(f), \mathrm{lm}(f)$ the leading term, leading coefficient and leading monomial of f respectively. The word "leading" is used in the context of the ordering $<$.*

We define $\deg_{x_i}(f)$ as the highest power of x_i, that appears in the monomials of f. We use $\deg(f)$ for the total degree of f, i.e. $\deg(f) = \sum_{i=1}^{n} \deg_{x_i}(lt(f))$, where the leading term is taken with respect to the graduated lexicographic ordering with a constant weight. As an example we take $f = x_1x_2^3 + x_1^2$. Then $\deg_{x_1}(f) = 2$ and $\deg(f) = 4$.

Definition 21. *Let K be a field. An admissible ordering $<$ on $[X]$ induces a partial ordering on $K[x_1, \ldots, x_n]$ as follows. Let $f, g \in K[x_1, \ldots, x_n]$.*

$$\begin{aligned} f < g \ &\textit{if } f = 0 \textit{ and } g \neq 0 \\ &\textit{or if } \mathrm{lt}(f) < \mathrm{lt}(g) \\ &\textit{or if } \mathrm{lt}(f) = \mathrm{lt}(g) \textit{ and } \mathrm{lc}(f) < \mathrm{lc}(g) \\ &\textit{or if } \mathrm{lm}(f) = \mathrm{lm}(g) \textit{ and } (f - \mathrm{lm}(f)) < (g - \mathrm{lm}(g)) \end{aligned}$$

We can use the third condition for $f < g$ if there is some natural, possibly partial ordering on $\mathbb{F}$, clear from the context.

3.2 Gröbner bases

At this point we have a sufficient background on orderings, to be able to introduce the Gröbner bases. The definitions, propositions, algorithms and some proofs in this section are based on those from the Chapter 6 in [SBff11].

Definition 22 (Gröbner basis)**.** *Fix an admissible ordering $<$ on $[X]$. Let $G = \{g_1, \ldots, g_m\} \subset \mathbb{F}[x_1, \ldots, x_n]$. G is a Gröbner basis if the reduction modulo G is unique in the following sense: Let $f \in \mathbb{F}[x_1, \ldots, x_n]$. We set $f' := f$. If there is such $g_j \in G$, that reduces f', i.e. $f' \mod g_j \neq f'$, we set $f' := f' \mod g_j$ and we repeat the process. If no $g_j \in G$ reduces f', we set $f \mod G := f'$. The reduction is unique if we always get the same result no matter which sequence of g_j's we use in the process.*

Note, that the reductions modulo g_j depend on the ordering $<$, because the $\mathrm{lt}(g_j)$ depends on it.

Definition 23 (s-polynomial)**.** *Let $f, f' \in \mathbb{F}[x_1, \ldots, x_n]$. We define*

$$spol(f, f') := \mathrm{lcm}(\mathrm{lt}(f), \mathrm{lt}(f')) \mod f - \mathrm{lcm}(\mathrm{lt}(f), \mathrm{lt}(f')) \mod f' \ .$$

Let F be a finite set of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. There are algorithms, that compute a Gröbner basis of $\langle F \rangle \subset \mathbb{F}[x_1, \ldots, x_n]$. One example is the Buchberger algorithm.

Algorithm 2 Buchberger Algorithm

Input: $F \subset \mathbb{F}[x_1, \ldots, x_n]$
Output: $G \subset \mathbb{F}[x_1, \ldots, x_n]$, such that $\langle F \rangle = \langle G \rangle$ and G is a Gröbner basis
  while $\exists (f, f') \in F : spol(f, f') \mod F \neq 0$ **do**
    $F := F \cup \{spol(f, f') \mod F\}$
  end while
  set $G := F$
  return G

Proposition 24. *Buchberger algorithm works.*

See [SBff11] or [Win96] for the proof.

Definition 25. *A Gröbner basis G is called reduced if for all $g \in G$ it holds*

$$g \mod G \setminus \{g\} = g \ .$$

A Gröbner basis G is called normed if for all $g \in G$ it holds

$$\mathrm{lc}(g) = 1 \ .$$

The Buchberger algorithm does not generally produce a reduced normed Gröbner basis. However, once we have some Gröbner basis, it is easy to reduce it as we can see in Algorithm 3.

A reduced Gröbner basis can be transformed into a normed one, by simply dividing each element of the basis by its leading coefficient.

Proposition 26. *Let I be an ideal in $\mathbb{F}[x_1, \ldots, x_n]$. Then there exists a unique normed reduced Gröbner basis G, such that $\langle G \rangle = I$.*

See [Win96] for the proof of this proposition.

Algorithm 3 Gröbner basis reduction

Input: $G \subset \mathbb{F}[x_1, \ldots, x_n]$ a Gröbner basis
Output: $\tilde{G} \subset \mathbb{F}[x_1, \ldots, x_n]$ a reduced Gröbner basis

```
while ∃ (f, f') ∈ G such, that f mod f' = g, f ≠ g do
    if lt(f') | lt(f) then
        set G := G \ {f}
    else
        set G := G \ {f} ∪ {g}
    end if
end while
set G̃ := G
return G̃
```

3.3 Zero-dimensional ideals

Definition 27. *Let K be a field and $I \subsetneq K[x_1, \ldots, x_n]$ an ideal. Then I is called zero-dimensional if*

$$\dim_K (K[x_1, \ldots, x_n]/I) < \infty \ .$$

Proposition 28. *Let $I \subsetneq K[x_1, \ldots, x_n]$ be an ideal. Let $G \subset I$ be the Gröbner basis of I. Then*

$$\dim_K (K[x_1, \ldots, x_n]/I) = |\{t \mid t \text{ is a term in } K[x_1, \ldots, x_n],\ t \mod G = t\}| \ .$$

Proof. It is easy to see, that all the terms of $K[x_1, \ldots, x_n]$ irreducible by G generate $K[x_1, \ldots, x_n]/I$ and they are K-linearly independent. It follows, that their number is the dimension of $K[x_1, \ldots, x_n]/I$ over K. □

Note, that if $\mathbb{F}$ is a finite field, $\mathbb{F} = \mathbb{F}_q$ then there is a natural mapping

$$\begin{array}{rccc} \pi: & \mathbb{F}[x_1, \ldots, x_n] & \longrightarrow & \mathbb{F}[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle \\ & f & \longmapsto & f \mod \langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle \end{array}$$

π induces an equivalence on $\mathbb{F}[x_1, \ldots, x_n]$, $f \sim f'$ on $\mathbb{F}[x_1, \ldots, x_n]$ if $\pi(f) = \pi(f')$. It follows that for all $a \in \mathbb{F}^n$ it holds $f(a) = f'(a)$.

Instead of working with $\mathbb{F}[x_1, \ldots, x_n]$ it is convenient to consider only the ring $\mathbb{F}[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle$. As this ring has a finite dimension over $\mathbb{F}$, so do all of its factor rings. We shall abuse the definition of a zero-dimensional ideal here and in the case of finite fields we say, that I, an ideal of $\mathbb{F}[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle$ is zero-dimensional if there exists $\nu \in \mathbb{N}$, $\nu < q - 1$ such that the $\deg_{x_i}$ of the terms in $\mathbb{F}[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle$ irreducible by I is bounded by ν for $i = 1, \ldots, n$. The reason, why we define it this way is that, in a general field, an ideal I is zero-dimensional if all the I-irreducible terms have a bounded degree in each variable. Each is bounded by some $f \in I$. When we use the expression for finite fields, we want to say, that all the degrees of terms are bounded by I, not by $\langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle$.

The following proposition appears in [CLO97] in a version for the polynomial rings over complex numbers. We state the same proposition for polynomial rings over finite fields.

Proposition 29. *Let $I \subset \mathbb{F}[x_1, \ldots, x_n]$ be an ideal. Let $V \subset \mathbb{F}^n$ be the algebraic set of I, i.e. $V := V(I) = \{r \in \mathbb{F}^n \mid f(r) = 0,\ \forall f \in I\}$. Then*

$$|V| \leq \dim_{\mathbb{F}} (\mathbb{F}[x_1, \ldots, x_n]/I) \ .$$

Proof. This proof is analogous to the proof of proposition's variant, presented in [CLO97]. We will show, that given a set $V = \{r^{(1)}, \ldots, r^{(k)}\} \subseteq \mathbb{F}^n$, $V = V(I)$, for some I, an ideal of $\mathbb{F}[x_1, \ldots, x_n]$, we can find k $\mathbb{F}$-linearly independent polynomials in $\mathbb{F}[x_1, \ldots, x_n]/I$. Then it will follow, that

$$|V| \leq \dim_{\mathbb{F}} (\mathbb{F}[x_1, \ldots, x_n]/I) \ .$$

First we show, that given $V = \{r^{(1)}, \ldots, r^{(k)}\} \subseteq \mathbb{F}^n$, we can find $f_1 \in \mathbb{F}[x_1, \ldots, x_n]$ that satisfies

$$\begin{aligned} f_1(r^{(1)}) &= 1 \ , \\ f_1(r^{(i)}) &= 0,\ i = 2 \ldots, k \ . \end{aligned}$$

For $i \neq 1$ it holds $r^{(i)} \neq r^{(1)}$, therefore we can find an $l = l(i) \in \{1, \ldots, n\}$, such that $r^{(i)}_{l(i)} \neq r^{(1)}_{l(i)}$. For $i = 2 \ldots, k$ we set

$$b_i := r^{(i)}_{l(i)} \text{ and } h_i := \frac{x_{l(i)} - b_i}{r^{(1)}_{l(i)} - b_i} \ .$$

We have $h_i(r^{(1)}) = 1$ and $h_i(r^{(i)}) = 0$. Now we set $\tilde{f}_1 := \prod_{i=2}^{k} h_i \in R$. We get

$$\begin{aligned} \tilde{f}_1(r^{(1)}) &= \prod_{i=2}^{k} h_i(r^{(1)}) = \prod_{i=2}^{k} 1 = 1 \ , \\ \tilde{f}_1(r^{(i)}) &= h_i(r^{(i)}) \prod_{j=2,\ j\neq i}^{k} h_j(r^{(i)}) = 0,\ i = 2 \ldots, k \ . \end{aligned}$$

Analogously, we can find polynomials $\tilde{f}_2, \ldots, \tilde{f}_k \in \mathbb{F}[x_1, \ldots, x_n]$, such that

$$\begin{aligned} \tilde{f}_j(r^{(j)}) &= 1,\ j = 2, \ldots, k, \ , \\ \tilde{f}_j(r^{(i)}) &= 0,\ j = 2, \ldots, k,\ i = 1, \ldots, j-1, j+1, \ldots, k \ . \end{aligned}$$

We set $f_j := \tilde{f}_j + I \in \mathbb{F}[x_1, \ldots, x_n]/I$, $j = 1, \ldots, k$. Now we need to show, that polynomials $f_1, \ldots, f_k \in \mathbb{F}[x_1, \ldots, x_n]/I$ are $\mathbb{F}$-linearly independent. Suppose, that for $a_i \in \mathbb{F}$, $i = 1, \ldots, k$ it holds $\sum_{i=1}^{k} a_i f_i = 0 \in \mathbb{F}[x_1, \ldots, x_n]/I$. It follows, that if we set $g = \sum_{i=1}^{k} a_i \tilde{f}_i$, we get $g \in I$, hence for $j = 1, \ldots, k$ we have $g(r^{(j)}) = 0$. On the other hand we get

$$g(r^{(j)}) = \sum_{i=1}^{k} a_i \tilde{f}_i(r^{(j)}) = 0 + a_j \tilde{f}_j(r^{(j)}) = a_j \ .$$

Hence $a_j = 0$ for $j = 1, \ldots, k$ and we conclude, that any $\mathbb{F}$-linear combination of f_is yielding zero must be trivial, i.e. $f_1, \ldots, f_k \in \mathbb{F}[x_1, \ldots, x_n]/I$ are $\mathbb{F}$-linearly independent. □

4. Polly Cracker

In 1994 Neal Koblitz et al. presented a general outline for a construction of a public-key cryptosystems based on NP-hard problems in [FK94]. As an example, they described a cryptosystem based on the ideal membership problem and named it Polly Cracker. A whole family of cryptosystems based on this construction has been developed over the following years. These have played a critical role in the development of homomorphic encryption theory, mostly serving as a base stone on which cryptographers built more sophisticated systems. For instance, Craig Gentry's fully homomorphic encryption system that uses lattices [Gen09] is based on Polly Cracker.

In this chapter we denote $\mathcal{R} = \mathbb{F}[x_1, ..., x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n\rangle$.

Algorithm 4 describes one particular version of Polly Cracker. In this cryptosystem, the set of messages is $\mathcal{P} = \mathbb{F}$, the set of ciphertexts is $\mathcal{C} = \mathcal{R}$ and the keys $K \in \mathcal{K}$ are pairs (SK, PK), where the secret key $SK = s$ is a vector in $\mathbb{F}^n$ and the public key $PK = \{f_1, \ldots, f_k\}$ is a set of polynomials in $\mathcal{R}$.

Proposition 30. *Algorithm 4 describes a probabilistic cryptosystem.*

Proof. Let us check the four conditions from the definition of a cryptosystem.

1. $\mathcal{P} = \mathbb{F}$ is a finite set.
2. $\mathcal{C} = \mathcal{R}$ is a finite set, because the degree of polynomials is bounded by $q-1$ in each variable, as explained in Section 3.3.
3. The key-space $\mathcal{K} = \{(s, f_1, ..., f_k) \mid s \in \mathbb{F}^n,\ \ f_i \in \mathbb{F}[x_1, ..., x_n],\ \deg(f_i) \leq \nu,\ f_j(s) = 0,\ j = 1, \ldots, k\}$ is a finite set of possible keys.
4. Let $m \in \mathbb{F}$, $K = (SK, PK) = (s, f_1, ..., f_k) \in \mathcal{K}$. For all the choices of I we have

$$\mathrm{Dec}_{SK}(\mathrm{Enc}_{PK}(m)) = \left(m + \sum_{j \in I} f_j\right)(s) = m + \sum_{j \in I} f_j(s) = m.$$

   As the key K and the message m were chosen arbitrarily, the Condition 4 of the Definition 1 is satisfied.

The choice of I in the function *Encrypt* is non-deterministic, i.e. there are different ways to encrypt one message and the cryptosystem is probabilistic, which concludes the proof. □

Proposition 31. *The cryptosystem described by Algorithm 4 is additively and multiplicatively homomorphic.*

Proof. Fix $(SK, PK) = (s, f_1, ..., f_k) \in \mathcal{K}$ and $c_1, c_2 \in \mathbb{F}[x_1, ..., x_n]$. We need to show that $\mathrm{Dec}_{SK}(c_1) + \mathrm{Dec}_{SK}(c_2) = \mathrm{Dec}_{SK}(add(c_1, c_2))$ and $\mathrm{Dec}_{SK}(c_1) \cdot$

Algorithm 4 Polly Cracker

SETUP

Input: $n, k, q, \nu \in \mathbb{N},\ q$ prime power, $\nu < q - 1$

Output: $(SK, PK),\ SK \in \mathbb{F}^n, PK \subset \mathcal{R}$

set $\mathbb{F} := \mathbb{F}_q$

choose $s = (s_1, ..., s_n) \leftarrow \mathbb{F}^n$

for $j = 1 \rightarrow k$ **do**

choose $f_j \leftarrow \mathbb{F}[x_1, \ldots, x_n]$ s.t. $\deg(f) \leq \nu,\ f_j(s) = 0$

end for

set the secret key $SK := s$

set the public key $PK := \{f_1, ..., f_k\}$

return the key (SK, PK)

ENCRYPT

Input: $m \in \mathbb{F}, PK = \{f_1 \ldots, f_k\}$

Output: $c \in \mathbb{F}[x_1, \ldots, x_n]$

select $I \subseteq \{1, .., k\}$ uniformly at random

set the ciphertext $c := m + \sum_{j \in I} f_j \in \mathcal{R}$

return c

DECRYPT

Input: $c \in \mathbb{F}[x_1, \ldots, x_n], SK = s \in \mathbb{F}^n$

Output: $m \in \mathbb{F}$

set $m := c(s)$

return m

ADD

Input: $c_1, c_2 \in \mathcal{R}$

Output: $c \in \mathcal{R}$

set $c := c_1 + c_2 \in \mathcal{R}$

return c

MULT

Input: $c_1, c_2 \in \mathcal{R}$

Output: $c \in \mathcal{R}$

set $c := c_1 \cdot c_2 \in \mathcal{R}$

return c

$\mathrm{Dec}_{SK}(c_2) = \mathrm{Dec}_{SK}(mult(c_1, c_2))$. We have

$$\begin{aligned}
\mathrm{Dec}_{SK}(c_1) + \mathrm{Dec}_{SK}(c_2) &= c_1(s) + c_2(s) \\
\mathrm{Dec}_{SK}(add(c_1, c_2)) &= \mathrm{Dec}_{SK}(c_1 + c_2) = (c_1 + c_2)(s) = c_1(s) + c_2(s) \\
\mathrm{Dec}_{SK}(c_1) \cdot \mathrm{Dec}_{SK}(c_2) &= c_1(s) \cdot c_2(s) \\
\mathrm{Dec}_{SK}(mult(c_1, c_2)) &= \mathrm{Dec}_{SK}(c_1 \cdot c_2) = (c_1 \cdot c_2)(s) = c_1(s) \cdot c_2(s) \quad .
\end{aligned}$$

The proof could also be formulated in the following way: The evaluation of polynomials in a point $s \in \mathbb{F}^n$ is a ring homomorphism:

$$\begin{aligned}
\varphi_s : \quad \mathbb{F}[x_1, \ldots, x_n] &\longrightarrow \mathbb{F} \\
f &\longmapsto f(s) \quad .
\end{aligned}$$

$\langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$ is an ideal of $\mathbb{F}[x_1, \ldots, x_n]$. By the fundamental theorem on homomorphisms it follows, that φ_s is a ring homomorphism of $\mathcal{R}$ and $\mathbb{F}$. □

This cryptosystem is an example the case when $e_K(\mathcal{P}) \subsetneq \mathcal{C}$. None of the ciphertexts $c \in \mathcal{C}$, such that $\deg(c) > \nu$, belongs to Enc_{PK}, but all decrypt to some $Dec_{SK}(c) = m \in \mathbb{F}$.

We have a fully homomorphic cryptosystem. We can do an unlimited number of additions and multiplications. However, the size of a ciphertext grows linearly with the number of multiplications. This property is not practical. Fortunately, we work with the ring $\mathcal{R}$, which is finite, so after about $\frac{q}{\nu}$ multiplications the ciphertexts stop growing. Note that the use of the ring $\mathbb{F}[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$ instead of $\mathbb{F}[x_1, \ldots, x_n]$ does not affect the decryption as $s_i^q = s_i$:

$$\mathrm{Dec}_{SK}(c \mod \langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle) = c(s) - q_i(s)(x_i^q - x_i)(s) = c(s) \quad .$$

Nevertheless, the size of a random polynomial in $\mathcal{R}$ in bits is $O(q^n)$. (The size of a random polynomial in $\mathcal{R}$ is $\log_2$(number of polynomials in $\mathcal{R}$) $= \log_2\left(q^{q^n}\right)$.) We need to keep the number of variables very low if we want the size of ciphertexts to stay reasonable.

Let us have a look at the security of the system. In the following we denote $V(f_1, \ldots, f_k) \subset \mathbb{F}^n$ the algebraic set of polynomials $f_1, \ldots, f_k \in \mathcal{R}$:

$$V(f_1, \ldots, f_k) = \{\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n \mid f_j(\mathbf{a}) = 0, j = 1, \ldots, m\} \quad .$$

Lemma 32. *Let* $\mathbf{a} \in V(f_1, \ldots, f_k)$. *Then* $\mathrm{Dec}_{\mathbf{a}} = \mathrm{Dec}_{SK}$.

This lemma follows straight from the definition of $V(f_1, \ldots, f_k)$. We see, that if there is more than one element in $V(f_1, \ldots, f_k)$, the attacker does not actually need to find s to be able to decrypt messages. That is a security flaw of the system.

Proposition 33. *Let* F *be the Gröbner basis of the ideal* $\langle f_1, \ldots, f_k \rangle$. *Let* $c \in \mathcal{R}$ *be such polynomial, that comes from a finite number of multiplications and additions of fresh encryptions of messages in* $\mathbb{F}$. *Then* c *decrypts to* 0 *if and only if* $c \mod F = 0$.

Proof. First we show the implication

$$c \mod F = 0 \Rightarrow \mathrm{Dec}_{SK}(c) = 0 \ .$$

Let $c \mod F = 0$. We denote $F = \{f'_1, \ldots, f'_{k'}\}$. Hence $c = \sum_{l=1}^{k'} q_l \cdot f'_l$ and

$$\mathrm{Dec}_{SK}(c) = \sum_{l=1}^{k'} q_l(s) \cdot f'_l(s) = 0 \ .$$

Now we show, that if c is a fresh encryption or if $c = c_1 \cdot c_2$ and c_1, c_2 are fresh encryptions, then the proposition holds. The rest of the proof follows by induction.

If c is a fresh encryption, then there exists such $I \subseteq \{1, \ldots, k\}$ that

$$c = \sum_{j \in I} f_j + m \ .$$

Hence $c \mod F = 0$ if and only if $\mathrm{Dec}_{SK}(c) = m = 0$. Now let $I_1, I_2 \subseteq \{1, \ldots, k\}$, such that

$$\begin{aligned} c = c_1 \cdot c_2 &= \left(\sum_{j \in I_1} f_j + m_1\right) \cdot \left(\sum_{j \in I_2} f_j + m_2\right) \\ &= \sum_{j \in I_1} f_j \cdot \sum_{j \in I_2} f_j + \sum_{j \in I_1} f_j \cdot m_2 + \sum_{j \in I_2} f_j \cdot m_1 + m_1 \cdot m_2 \ . \end{aligned}$$

Hence $c \mod F = 0$ if and only if $\mathrm{Dec}_{SK}(c) = m_1 \cdot m_2 = 0$.

Note that to prove the first implication we did not need to put any restriction on the origin of c. □

Corollary 34. *If an attacker knows F, the Gröbner basis of $\langle f_1, \ldots, f_k \rangle$, then for a ciphertext $c \in \mathcal{C}$ (not necessarily a fresh encryption), he can calculate $m = \mathrm{Dec}_{SK}(c) = c \mod F$.*

Proof. Let $c \in \mathcal{R}$ a ciphertext, such that $\mathrm{Dec}_{SK}(c) = m$. We denote $c^* = c - m \in \mathcal{R}$ is an encryption of 0. The decryption is linear, hence we have $\mathrm{Dec}_{SK}(c^* + m) = \mathrm{Dec}_{SK}(c^*) + m$. Together we get

$$\begin{aligned} c \mod F = c^* \mod F + m \mod F &= \mathrm{Dec}_{SK}(c^*) + m \\ &= \mathrm{Dec}_{SK}(c^* + m) = \mathrm{Dec}_{SK}(c) \ , \end{aligned}$$

which concludes the proof. □

The attacker does need to calculate F, the Gröbner basis of $\langle f_1, \ldots, f_k \rangle$ first. This calculation does have up to double-exponential complexity in the number of variables, but as we have shown earlier, the number of variables needs to be very small if we want to have a reasonable size of ciphertexts.

5. Symmetric Polly Cracker - version 1

In this section we propose a new fully homomorphic symmetric encryption scheme inspired by Polly Cracker and the scheme described by Armknecht et. al. in Section 2.2. The scheme is called **SymPC**. The first general version is SymPC1.

The Chapters 6 and 7 explore some of the scheme's weaknesses and propose optimized parameter settings. On this knowledge we build the second version SymPC2 in Chapter 8.

We shall use the notation

$$\mathcal{L} := \mathbb{F}[x_1, \ldots, x_n]/\langle g_1, \ldots, g_n \rangle$$

In SymPC1,described by Algorithm 5, the set of messages is $\mathcal{P} = \mathbb{F}$, the set of ciphertexts is $\mathcal{C} = LL \subset \mathbb{F}[x_1, \ldots, x_n]$ and the keys $K \in \mathcal{K}$ are pairs (SK, MK), where the secret key $SK = s$ is a vector in $\mathbb{F}^n$ and $MK = \{g_1, \ldots, g_n\}$, the multiplication key, is a set of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. It is a special kind of key, that is only used in the multiplication of ciphertexts. It provides information about the ring of ciphertexts $\mathcal{C}$. It is public, but one cannot encrypt messages with it. That is why we do not call it a public key.

Proposition 35. *Algorithm 5 describes a probabilistic cryptosystem.*

Proof. Let us check the four conditions from the definition of a cryptosystem.

1. $\mathcal{P} = \mathbb{F}$ is a finite set.
2. $\mathcal{C} = \mathcal{L} \subset \mathbb{F}[x_1, ..., x_n]$ is a finite set because the degree of polynomials is bounded by ν. This is further explained in Propositions 37 and 38.
3. The key-space $\mathcal{K} = \{(s, g_1, ..., g_n) \mid s \in \mathbb{F}^n,\ g_i \in \mathbb{F}[x_1, \ldots, x_n],\ g_i(s) = 0,\ i = 1, \ldots, n\}$, is a finite set of possible keys.
4. Let $m \in \mathbb{F}$, $K = (SK, MK) \in \mathcal{K}$. We have

   $$\mathrm{Dec}_K(\mathrm{Enc}_K(m)) = (\mathrm{Enc}_K(m))(s) = (f - f(SK) + m)(SK) = m.$$

   As the key K and the message m were chosen arbitrarily, the Condition 4 of Definition 1 is satisfied.

The random choice of f in the function Encrypt makes the encryption non-deterministic, hence the cryptosystem is probabilistic, which concludes the proof. □

Proposition 36. *The cryptosystem SymPC1 is additively homomorphic.*

Algorithm 5 SymPC1

SETUP

Input: $n, \nu, q \in \mathbb{N}, \nu < q-1,\ q$ is a prime power

Output: $SK \in \mathbb{F}^n, MK \subset \mathbb{F}[x_1, \ldots, x_n]$

set $\mathbb{F} := \mathbb{F}_q$

choose $s = (s_1, ..., s_n) \leftarrow \mathbb{F}^n$

for $i = 1 \to n$ **do**

choose $\tilde{g}_i \leftarrow \mathbb{F}[x_1, \ldots, x_n]$, $\deg(\tilde{g}_i) < \nu$

set $g_i := (x_i - s_i) \cdot (x_i^\nu + \tilde{g}_i)$

end for

set the secret key $SK := s$

set the multiplication key $MK := \{g_1, \ldots, g_n\}$

return (SK, MK)

ENCRYPT

Input: message $m \in \mathbb{F}, SK \in \mathbb{F}^n$

Output: $c \in \mathcal{L}$

choose $f \leftarrow \mathbb{F}[x_1, \ldots, x_n]$, $\deg_{x_i}(f) \leq \nu$, $i = 1, \ldots, n$

set $c := f - f\,(SK) + m \in \mathcal{L}$

return c

DECRYPT

Input: $c \in \mathcal{L},\ SK \in \mathbb{F}^n$

Output: $m \in \mathbb{F}$

set $m := c\,(SK)$

return m

ADD

Input: $c_1, c_2 \in \mathcal{L}$

Output: $c \in \mathcal{L}$

set $c := c_1 + c_2 \in \mathcal{L}$

return c

MULT_{MK}

Input: $c_1, c_2 \in \mathcal{L}$

Output: $c \in \mathcal{L}$

set $c := c_1 \cdot c_2 \mod \langle g_1, \ldots, g_n \rangle$

return c

Proof. Fix $K \in \mathcal{K}$ and $c_1, c_2 \in \mathcal{L}$. We need to show that $\mathrm{Dec}_K(c_1) + \mathrm{Dec}_K(c_2) = \mathrm{Dec}_K(add(c_1, c_2))$. We have

$$\mathrm{Dec}_K(c_1) + \mathrm{Dec}_K(c_2) = c_1(s) + c_2(s)$$

$$\mathrm{Dec}_K(add(c_1, c_2)) = \mathrm{Dec}_K(c_1 + c_2) = (c_1 + c_2)(s) = c_1(s) + c_2(s).$$

□

Proposition 37. *$\{g_1, ..., g_n\}$ is a reduced Gröbner basis of $\langle g_1, ..., g_n \rangle$.*

Proof.

$$\begin{aligned} g_i &= (x_i - s_i) \cdot (x_i^\nu + \tilde{g}_i) \quad , \text{ where } \deg(\tilde{g}_i) < \nu \\ &= x_i^{\nu+1} - s_i x_i^\nu + x_i \tilde{g}_i - s_i \tilde{g}_i \end{aligned}$$

We set $F := \{g_1, \ldots, g_n\} \subset \mathbb{F}[x_1, \ldots, x_n]$. If we show, that for all i, j $i \neq j$ the s-polynomial $spol(g_i, g_j) = 0 \mod F$, we get, that Buchberger Algorithm (Algorithm 2) outputs $G = F$, i.e. $\{g_1, \ldots, g_n\}$ is a Gröbner basis.

We have

$$\begin{aligned} spol(g_i, g_j) &= \mathrm{lcm}(\mathrm{lt}(g_i), \mathrm{lt}(g_j)) \mod g_i - \mathrm{lcm}(\mathrm{lt}(g_i), \mathrm{lt}(g_j)) \mod g_j \\ &= x_i^{\nu+1} \cdot x_j^{\nu+1} \mod g_i - x_i^{\nu+1} \cdot x_j^{\nu+1} \mod g_j \\ &= (s_i x_i^\nu - x_i \tilde{g}_i + s_i \tilde{g}_i) \cdot x_j^{\nu+1} - \left(s_j x_j^\nu - x_j \tilde{g}_j + s_j \tilde{g}_j\right) \cdot x_i^{\nu+1} \\ spol(g_i, g_j) \mod F &= ((s_i x_i^\nu - x_i \tilde{g}_i + s_i \tilde{g}_i) \cdot x_j^{\nu+1} \\ &\quad - \left(s_j x_j^\nu - x_j \tilde{g}_j + s_j \tilde{g}_j\right) \cdot x_i^{\nu+1}) \mod F \\ &= ((s_i x_i^\nu - x_i \tilde{g}_i + s_i \tilde{g}_i) \cdot x_j^{\nu+1} \mod g_j \\ &\quad - \left(s_j x_j^\nu - x_j \tilde{g}_j + s_j \tilde{g}_j\right) \cdot x_i^{\nu+1} \mod g_i) \mod F \\ &= (s_i x_i^\nu - x_i \tilde{g}_i + s_i \tilde{g}_i)\left(s_j x_j^\nu - x_j \tilde{g}_j + s_j \tilde{g}_j\right) \\ &\quad - \left(s_j x_j^\nu - x_j \tilde{g}_j + s_j \tilde{g}_j\right)(s_i x_i^\nu - x_i \tilde{g}_i + s_i \tilde{g}_i) \mod F \\ &= 0 \mod F = 0 \end{aligned}$$

We see, that $\{g_1, \ldots, g_n\}$ is a Gröbner basis. For all i, j distinct, we have $g_i \mod g_j = g_i$, so $\{g_1, \ldots, g_n\}$ is a reduced Gröbner basis. Finally, all g_is are monic, so $\{g_1, \ldots, g_n\}$ is a normed reduced Gröbner basis. □

Proposition 38. *Let $\{g_1, \ldots, g_n\}$ be defined by in Algorithm 5. Then all the polynomials in $\mathcal{L} = \mathbb{F}[x_1, ..., x_n]/\langle g_1, ..., g_n \rangle$ have a degree at most $\nu \cdot n$. $\langle g_1, ..., g_n \rangle$ is a zero-dimensional ideal and*

$$\dim_{\mathbb{F}}(\mathbb{F}[x_1, ..., x_n]/\langle g_1, ..., g_n \rangle) = (\nu + 1)^n \quad .$$

Proof. We see, that $\deg_{x_i}(g_i) = \nu + 1$, $i = 1, \ldots, n$. The proposition is a direct consequence of the Proposition 28 and its following remark. □

Corollary 39. *Let $V(g_1, \ldots, g_n) = \{r \in \mathbb{F}^n \mid g_i(r) = 0,\ i = 1, \ldots, n\}$ be the algebraic set of the zero-dimensional ideal generated by $g_1, ..., g_n$. Then*

$$|V(g_1, \ldots, g_n)| \leq (\nu + 1)^n \quad .$$

Proof. Follows straight from Propositions 29 and 38. □

Proposition 40. *The cryptosystem SymPC1 is multiplicatively homomorphic.*

Proof. Fix $K \in \mathcal{K}$ and $c_1, c_2 \in \mathcal{L}$. We need to show that $\mathrm{Dec}_K(c_1) \cdot \mathrm{Dec}_K(c_2) = \mathrm{Dec}_K(mult_{MK}(c_1, c_2))$. First, note that according to the Proposition 37, the definition of $mult_{MK}$ makes sense and $mult_{MK}(c_1, c_2)$ is indeed in $\mathcal{L}$. We have

$$\begin{aligned} \mathrm{Dec}_K(c_1) \cdot \mathrm{Dec}_K(c_2) &= c_1(s) \cdot c_2(s) \\ \mathrm{Dec}_K(mult_{MK}(c_1, c_2)) &= \mathrm{Dec}_K(c_1 \cdot c_2 \mod \langle g_1, ..., g_n \rangle) \\ &= (c_1 \cdot c_2 \mod \langle g_1, ..., g_n \rangle)(s) = c_1(s) \cdot c_2(s) \ , \end{aligned}$$

because $c_1 \cdot c_2 \mod \langle g_1, ..., g_n \rangle = c_1 \cdot c_2 - \sum_{i=1}^{n} q_i g_i$ for some $q_i \in \mathbb{F}[x_1, ..., x_n]$ and $(c_1 \cdot c_2 \mod \langle g_1, ..., g_n \rangle)(s) = (c_1 \cdot c_2)(s) - \sum_{i=1}^{n} q_i(s)\, g_i(s) = (c_1 \cdot c_2)(s)$, as $g_i(s) = 0$ for all $i = 1, \ldots, n$. Therefore,

$$\mathrm{Dec}_K(mult_{MK}(c_1, c_2)) = \mathrm{Dec}_K(c_1) \cdot \mathrm{Dec}_K(c_2) \ .$$

□

5.1 Complexity of SymPC1

In this section we evaluate the complexity of each function of SymPC1. We identify the most time-consuming parts and use this information to suggest a more efficient scheme in Chapter 8.

SETUP
Input: $n, \nu, q \in \mathbb{N}, \nu < q - 1,\ q$ is a prime power
Output: $SK \in \mathbb{F}^n, MK \subset \mathbb{F}[x_1, \ldots, x_n]$
set $\mathbb{F} := \mathbb{F}_q$
choose $s = (s_1, ..., s_n) \leftarrow \mathbb{F}^n$
for $i = 1 \to n$ **do**
choose $\tilde{g}_i \leftarrow \mathbb{F}[x_1, \ldots, x_n]$, $\deg(\tilde{g}_i) < \nu$
set $g_i := (x_i - s_i) \cdot (x_i^{\nu} + \tilde{g}_i)$
end for
set the secret key $SK := s$
set the multiplication key $MK := \{g_1, \ldots, g_n\}$
return (SK, MK)

In the Setup phase, the most complex operation is generation of the polynomials g_i. The algorithm generates n random polynomials $\tilde{g}_i$ of total degree less than ν. We denote β_i the number of coefficients of $\tilde{g}_i$. We have β_i is the number of terms of degree less than ν in $\mathbb{F}[x_1, \ldots, x_n]$, that is $\beta_i = O\left(\frac{\nu^n}{(n+1)!}\right)$ for $i = 1, \ldots, n$. For each $i = 1, \ldots, n$ the algorithm performs $\log_2 q \cdot \beta_i$ assignments of random bits to coefficients of $\tilde{g}_i$, $2 \cdot \beta_i$ multiplications and β_i additions of elements in $\mathbb{F}$. The overall complexity of the Setup phase is $O\left(\frac{\nu^n}{n!}\right)$ **operations in** $\mathbb{F}$.

ENCRYPT
Input: message $m \in \mathbb{F}, SK \in \mathbb{F}^n, \nu \in \mathbb{N}$
Output: $c \in \mathcal{L}$
choose $f \leftarrow \mathbb{F}[x_1, \ldots, x_n]$, $\deg_{x_i}(f) \leq \nu$
set $c = f - f(SK) + m \in \mathcal{L}$
return c
end

The most complex operation in Encrypt is the evaluation of f in $s = SK$. We denote α the number of coefficients of $f \in \mathcal{L}$. We have $\alpha = (\nu+1)^n$. The algorithm performs $\log_2 q \cdot \alpha$ assignments of random values to coefficients of f, α evaluations of monomials in $\mathcal{L}$. Each of these evaluations consists of at most $\deg(f) \leq \nu \cdot n$ multiplications in $\mathbb{F}$. Then it adds evaluations in the monomials. The overall complexity is $O(n \cdot (\nu+1)^{n+1})$ **operations in** $\mathbb{F}$. We calculated the complexity of a naive evaluation algorithm. Likely, it could be optimized, but we lower the complexity by a modification of the encryption in Chapter 8 instead.

DECRYPT
Input: $c \in \mathcal{L},\ SK \in \mathbb{F}^n$
Output: $m \in \mathbb{F}$
set $m := c(SK)$
return m

The complexity of this function is the same as the complexity of Encrypt, that is $O(n \cdot (\nu+1)^{n+1})$ **operations in** $\mathbb{F}$.

ADD
Input: $c_1, c_2 \in \mathcal{L}$
Output: $c \in \mathcal{L}$
set $c := c_1 + c_2 \in \mathcal{L}$
return c
end

The function performs α additions in $\mathbb{F}$, so the complexity is $O((\nu+1)^n)$ **operations in** $\mathbb{F}$.

MULT$_{MK}$
Input: $c_1, c_2 \in \mathcal{L}$
Output: $c \in \mathcal{L}$
set $c := c_1 \cdot c_2 \mod \langle g_1, \ldots, g_n \rangle$
return c

The function consists of two parts, multiplication and reduction. The first part is more complex, it involves α^2 multiplications in $\mathbb{F}$. The overall complexity is $O((\nu+1)^{2n})$ **operations in** $\mathbb{F}$.

6. Attacks on SymPC1

In this chapter we evaluate security against chosen-ciphertext (CCA), chosen-plaintext (CPA) and known-plaintext (KPA) attacks. Formal definitions of these may be found in [Sti95].

Proposition 41. *SymPC1 cryptosystem is not CCA secure.*

Proof. If an attacker can use an oracle, that decrypts ciphertexts, i.e.

$$c \xrightarrow{oracle} \mathrm{Dec}_{SK}(c) = c(s)$$

then he can just ask for the decryption of $(c_1, \ldots, c_n) = (x_1, \ldots x_n)$. As an answer he will receive $(s_1, \ldots, s_n)$, the points of secret key. Apparently, the system is not CCA-secure. □

Lemma 42. *For SymPC1 cryptosystem, the CPA-security is equivalent to KPA-security.*

Proof. CPA-security implies KPA-security in general. To proof the other implication we need to realize, that if an attacker has a known plaintext-ciphertext pair (m, c), where $\mathrm{Dec}_{SK}(c) = m$, he can obtain a pair (m', c') where m' is chosen arbitrarily and $c' = c - m + m'$. Then $\mathrm{Dec}_{SK}(c') = (m')$. From any plaintext-ciphertext pair, he can devise a chosen plaintext-ciphertext pair, so the cryptosystem needs to be CPA-secure to achieve the KPA-security. □

In the rest of the chapter we shall not differentiate between KPA and CPA attacks.

We say, that a cryptosystem is bounded CPA-secure if there exists an $m \in \mathbb{N}$, such that the cryptosystem is CPA-secure as long as no more than m plaintexts can be encrypted with the same key, i.e. the attacker cannot obtain more than m plaintext-ciphertext pairs.

Conjecture 43. *SymPC1 cryptosystem achieves a bounded CPA-security.*

There are two obvious approaches to breaking SymPC1 by a CPA attack. One is to try to calculate the Gröbner basis of the ideal

$$S = \{f \in \mathbb{F}[x_1, \ldots, x_n] \mid f(s) = 0\}$$

using $G = \{g_1, \ldots, g_n\}$ and $\{f_1, \ldots, f_m\}$, a set of encryptions of $0 \in \mathbb{F}$. The other approach consists of eliminating the algebraic set of G, hoping to end up only with the secret s.

Algorithm 6 Attack by computing the Gröbner basis

Input: $G \subseteq S \subset \mathbb{F}[x_1, \ldots, x_n], \{f_1, \ldots, f_m\} \in S$

Output: $F \subseteq \mathbb{F}[x_1, \ldots, x_n]$, such that $\langle F \rangle = \langle G, f_1, \ldots, f_m \rangle \subset S$ and F is a normed reduced Gröbner basis

```
set F := G
for l = 1 → m do
    set F the normed reduced Gröbner basis of {F, f_l}
    if F is of the form {x_1 − s_1, ..., x_n − s_n} then
        return F
    end if
end for
return F
```

6.1 Gröbner Basis approach

Here is an outline of an algorithm, that tries to calculate the Gröbner basis of S. Assume that an attacker obtains $\{f_1, \ldots, f_m\}$, encryptions of 0 and he knows the public multiplicative key $MK = G$. Algorithm 6 calculates the normed reduced Gröbner basis of the ideal generated by $\{g_1, \ldots, g_n, f_1, \ldots, f_m\} \subset S$. If $S = \langle g_1, \ldots, g_n, f_1, \ldots, f_m \rangle$ it has to output $\{x_1 - s_1, \ldots, x_n - s_n\}$, because the normed reduced Gröbner basis of an ideal is unique, as we stated in Proposition 26.

We see that if Algorithm 6 calculated the basis of S, we can easily read the secret key $s = (s_1, \ldots, s_n)$ from the output. The two important questions are:

What is the complexity of the algorithm?
What is the chance, that the algorithm actually calculated the basis of S?

The first question is difficult to answer, because it is even hard to tell, how long a calculation of a Gröbner basis takes in general. This is calculation of a Gröbner basis, given a "partial" Gröbner basis and further generators of an ideal. Intuitively, this may have a lower time-complexity than the general case.

The second question asks, what is the chance, that the ideal generated by G and $\{f_1, \ldots, f_m\}$ is equal to S. Here we have a more satisfying answer than the first question. We look at the ideals

$$\langle G \rangle = \langle F_0 \rangle \subseteq \langle F_1 \rangle \subseteq \cdots \subseteq \langle F_m \rangle \subseteq \cdots = S \ ,$$

where $F_0 = G = \{g_1, \ldots, g_n\}$ and for $j = 1, \ldots, m$, $F_j = \{g_1, \ldots, g_n, f_1, \ldots, f_j\}$.

In the Chapter 7 we show, that if the polynomials f_j from S are chosen uniformly at random and if we keep increasing m, the mean value of m at the point when we reach $\langle F_m \rangle = S$ is about $m = n$. This estimate is only valid for $|\mathbb{F}| = q$ large enough and under certain other conditions. Once we have proven this, we can just prevent the attacker from seeing more than the critical number of plaintext-ciphertext pairs. No matter what the complexity of the attack is, it will never work as the reduction of a ciphertext modulo $\langle F_{m-1} \rangle$ does not give us any information about the plaintext. For more details refer to Section 7.3.

6.2 Algebraic sets approach

The attack by Algorithm 7 finds the algebraic set of the ideal generated by G which is

$$V(G) = \{r \in \mathbb{F}^n \mid g(r) = 0,\ \forall g \in G\} \ .$$

Then it takes f_j's one by one and computes the intersection of the algebraic sets $V(G) \cap V(f_1) \cap \ldots \cap V(f_j)$.

Algorithm 7 Attack by computing the algebraic sets

Input: $G,\ G \subseteq S \subset \mathbb{F}[x_1, \ldots, x_n], \{f_1, \ldots, f_m\} \in S$
Output: $V,\ V \subset \mathbb{F}^n$, such that $V = V(G, f_1, \ldots, f_m)$

```
set V := ∅
for all r ∈ F^n do
    if g_i(r) = 0, i = 1, ..., n then
        set V := V ∪ {r}
    end if
end for
for l = 1 → m do
    set Ṽ := ∅
    for all r ∈ V do
        if f_l(r) = 0 then
            set Ṽ := Ṽ ∪ {r}
        end if
    end for
    set V := Ṽ
end for
return V
```

Since $G \subseteq S$ and $\{f_1, \ldots, f_m\} \subset S$, we see that $V(S) \subseteq V(G)$ and $V(S) \subseteq V(f_j)$, for $j = 1, \ldots, m$. It is obvious, that $V(S) = \{s\}$, hence

$$s \in V(G) \cap \bigcap_{j=1}^{m} V(f_j) = V(G, f_1, \ldots, f_m) \ .$$

We know, that Algorithm 7 outputs a non-empty set of candidates for the secret key s. Let V be the output of Algorithm 7. If V has only one element, we know, that it is the secret key s and the attack is successful. On the other hand, if the output set V contains a lot of elements, say $|V| \gg q$, then every ciphertext c may be decrypted in many ways. In particular, we know that $\mathrm{Dec}_{SK}(c) = c(s) \in c(V) = \{c(r) \mid r \in V\}$. $c(V) \subseteq \mathbb{F}$, its size grows with $|V|$ and we approach to the point when $c(V) = \mathbb{F}$. We see, that if $c(V)$ is large enough, we gain very little or no information about $\mathrm{Dec}_{SK}(c)$. This is further explained in Section 7.3.

The complexity of Algorithm 7 is $\mathcal{O}(q^n)$ evaluations in $\mathbb{F}[x_1, \ldots, x_n]$ and does not actually depend on m. It seems that, like it may be faster than Algorithm 6.

Note, that the two attacks are equivalent in certain way. The first one calculates Gröbner bases of ideals $F_0, \ldots, F_m$ and the other one calculates the algebraic sets of the very same ideals.

In particular, Algorithm 7 computes the secret key s if and only if $V(S) = \{s\}$ $= V(G, f_1, \ldots, f_m)$, i.e. it succeeds if and only if $\langle G, f_1, \ldots, f_m \rangle = F_m = S$. This is exactly the case when Algorithm 6 succeeds.

The success of both attacks may be avoided by bounding the maximum number of messages, that are allowed to be encrypted with one key. The adequate bound $m \in \mathbb{N}$ depends on the size of $\mathbb{F}$ and the number of variables. Suggestions on the parameter setting are given in Section 7.3.

7. Security of SymPC1

In the previous chapter we identified a threat imposed by two attacks. We also stated that these two attacks may or may not be successful depending on the circumstances. In this chapter we describe the circumstances under which the attacks succeed. Recall that the two attacks are equivalent on the matter of succeeding. Either both attacks succeed of both fail. We find it more transparent to deal with the algebraic set attack.

After description of the circumstances, we evaluate the likelihood of these occurring. This likelihood depends on the Setup parameters n, q, ν, the bound on the number of plaintexts allowed to be encrypted with the same key and the particular choice of the multiplication key $MK = \{g_1, \ldots, g_n\}$ in the Setup.

Let us recall and define notation used in this chapter:
The secret key $SK = s = (s_1, \ldots, s_n) \in \mathbb{F}^n$.
In this chapter only, we denote $R = \mathbb{F}[x_1, \ldots, x_n]$.
For $i = 1, \ldots, n$ $g_i \in R$ are chosen by Setup of SymPC1 as follows:

$$g_i := (x_i - s_i) \cdot (x_i^\nu + \tilde{g}_i) \quad ,$$

where $\tilde{g}_i$ is such that $\tilde{g}_i \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg(\tilde{g}_i) < \nu$. For $i = 1, \ldots, n$, the polynomials $\deg(g_i) = \nu + 1$ and $g_i(s) = 0$.

S and G are ideals in R defined as follows:

$$S = \langle x_1 - s_1, \ldots, x_n - s_n \rangle$$

$$G = \langle g_1, \ldots, g_n \rangle$$

We can see that $G \subset S$. In Chapter 5 we showed that $\{g_1, \ldots, g_n\}$ is a Gröbner basis of G and G is a zero-dimensional ideal in $\mathbb{F}[x_1, \ldots, x_n]$. We have the algebraic set $V(S) = \{s\}$,

$$V(G) = \{r \in \mathbb{F}^n \mid g_i(r) = 0, i = 1, \ldots, n\}$$

and $s \in V(G)$. We set $t := |V(G)|$. According to Proposition 29, $t \leq (\nu + 1)^n$. We number the elements of $V(G)$:

$$V(G) = \left\{r^{(1)}, \ldots, r^{(t)}\right\}, \; r^{(t)} = s$$

We denote $r^{(j)} = \left(r_1^{(j)}, \ldots, r_n^{(j)}\right), \; j = 1 \ldots, t.$

We would like to show that the mapping

$$\begin{array}{rccc} \varphi: & S/G & \longrightarrow & \mathbb{F}^{t-1} \times \{0\} \\ & f & \longmapsto & (f(r^{(1)}), \ldots, f(r^{(t)})) \end{array}$$

is a surjective homomorphism of vector spaces over $\mathbb{F}$. From this result we shall devise information on the distribution of zeros of polynomials in S/G.

7.1 Preliminary lemmas

Lemma 44. *The mapping*

$$\begin{array}{rccl} \varphi: & R/G & \longrightarrow & \mathbb{F}^t \\ & f & \longmapsto & (f(r^{(1)}),\ldots,f(r^{(t)})) \ , \end{array}$$

where $r^{(1)},\ldots,r^{(t)}$ are elements of $V(G)$, is a homomorphism of vector spaces over $\mathbb{F}$.

Proof. We have $\varphi(0) = (0,\ldots,0) \in \mathbb{F}^t$. Let $f, g \in R/G$, $a \in \mathbb{F}$. We have $f+g, a\cdot f \in R/G$,

$$\begin{aligned} \varphi(f+g) &= ((f+g)(r^{(1)}),\ldots,(f+g)(r^{(t)})) \\ &= (f(r^{(1)}),\ldots,f(r^{(t)})) + (g(r^{(1)}),\ldots,g(r^{(t)})) \\ &= \varphi(f) + \varphi(g) \\ \varphi(af) &= (af(r^{(1)}),\ldots,af(r^{(t)})) \\ &= a\cdot(f(r^{(1)}),\ldots,f(r^{(t)})) = a\cdot\varphi(f) \ . \end{aligned}$$

We have shown, that φ is a homomorphism of vector spaces over $\mathbb{F}$. □

The mapping φ is actually a ring homomorphism and the lemma could be proved by the fundamental theorem on homomorphism, but we decided to present only a weaker version of the lemma and a transparent proof for better clarity of the following propositions.

Corollary 45. *The mapping*

$$\begin{array}{rccl} \varphi: & S/G & \longrightarrow & \mathbb{F}^{t-1}\times\{0\} \\ & f & \longmapsto & (f(r^{(1)}),\ldots,f(r^{(t)})) \end{array}$$

is a homomorphism of vector spaces over $\mathbb{F}$.

Proof. This corollary is a direct consequence of the previous lemma and the fact, that for $f \in S$ it holds $f(r^{(t)}) = f(s) = 0$. □

Proposition 46. *The homomorphism $\varphi: S/G \longrightarrow \mathbb{F}^{t-1}\times\{0\}$ is surjective.*

Proof. Proof of this proposition is very similar to the proof of Proposition 29. However, we need to pay a special attention to the fact, that we work with the ideal S. To avoid any confusion we present the full proof.

Let $u_i = (0,\ldots,1,\ldots,0) \in \mathbb{F}^t$ be a vector with a 1 at the i-th position, $i = 1,\ldots,t-1$. We show that we can find $f \in S/G$, such that $\varphi(f) = u_i$. As i has been chosen arbitrarily and φ is a linear mapping, this will show, that φ is surjective.

The desired f needs to satisfy

$$\begin{aligned} &f(r^{(i)}) = 1 \ , \\ &f(r^{(j)}) = 0, \ j = 1\ldots,i-1,i+1,\ldots,t \ . \end{aligned}$$

For $j \neq i$ it holds $r^{(j)} \neq r^{(i)}$, therefore we can find an $l = l(j) \in \{1, \ldots, n\}$, such that $r^{(j)}_{l(j)} \neq r^{(i)}_{l(j)}$. For $j = 1 \ldots, i-1, i+1, \ldots, t$ we set

$$b_j := r^{(j)}_{l(j)} \text{ and } h_j := \frac{x_{l(j)} - b_j}{r^{(i)}_{l(j)} - b_j} \ .$$

We have $h_j(r^{(i)}) = 1$ and $h_j(r^{(j)}) = 0$. Now we set $\tilde{f} := \prod_{j=1,\ j\neq i}^{t} h_j \in R$. We have

$$\tilde{f}(r^{(i)}) = \prod_{j=1,\ j\neq i}^{t} h_j(r^{(i)}) = \prod_{j=1,\ j\neq i}^{t} 1 = 1 \ ,$$
$$\tilde{f}(r^{(j)}) = h_j(r^{(j)}) \cdot \prod_{k=1,\ k\neq i,j}^{t} h_k(r^{(j)}) = 0, \ j = 1 \ldots, i-1, i+1, \ldots, t \ .$$

We want to show that $\tilde{f} \in S$. We have $\tilde{f}(s) = \tilde{f}(r^{(t)}) = 0$, therefore $\tilde{f} \in S$.

Now we set $f := \tilde{f} + G \in S/G$. As $r^{(j)} \in V(G)$ for $j = 1, \ldots, t$, we get

$$f(r^{(i)}) = \tilde{f}(r^{(i)}) = 1 \ ,$$
$$f(r^{(j)}) = \tilde{f}(r^{(j)}) = 0, \ j = 1 \ldots, i-1, i+1, \ldots, t \ .$$

We have found $f \in S/G$, such that $\varphi(f) = u$ and the proof is concluded. □

Corollary 47.

$$S/G \Big/ \operatorname{Ker}(\varphi) \simeq \mathbb{F}^{t-1} \ ,$$

hence for every $u \in \mathbb{F}^{t-1}$ there exist $|\operatorname{Ker}(\varphi)|$ polynomials f in S/G, such that $\varphi(f) = u||0$.

Proof. This is a direct consequence of Proposition 46 and the First Isomorphism Theorem. □

Corollary 48. *Choose $f \in S/G$ uniformly at random. Then for all $u \in \mathbb{F}^{t-1} \times \{0\}$ it holds*

$$\Pr[\varphi(f) = u] = \frac{1}{|\mathbb{F}|^{t-1}} \ ,$$

i.e. $\varphi(f)$ is distributed uniformly over $\mathbb{F}^{t-1} \times \{0\}$. In particular, for $r \in V(G)$, $r \neq s$ it holds

$$\Pr[f(r) = 0] = \frac{1}{|\mathbb{F}|} = q^{-1} \ .$$

Furthermore, the values $f(r^{(1)}), \ldots, f(r^{(t-1)})$ are independent.

Proof. Follows straight from the previous corollary. □

7.2 Security evaluation

Let $f_1, \ldots, f_m$ be encryptions of 0, i.e. $f_1, \ldots, f_m$ are chosen from S/G uniformly at random. As we suggested in Chapter 6, we want to calculate the expected size of $V(G, f_1, \ldots, f_m)$. This will be the expected number of candidates for the secret key s. From the Corollary 48 we get, that $\varphi(f_1), \ldots, \varphi(f_m)$ are independent vectors, uniformly distributed over $\mathbb{F}^{t-1} \times \{0\}$. In particular, for $i = 1, \ldots, t-1$, the i-th coordinates of vectors $\varphi(f_l)$, these are $f_1(r^{(i)}), \ldots, f_m(r^{(i)})$, are independent, uniformly distributed elements of $\mathbb{F}$.

Let $r \in V(G) \setminus \{s\}$. Then

$$\Pr[r \in V(f_1, \ldots, f_m)] = \Pr[f_l(r) = 0,\ l = 1, \ldots, m] = \prod_{l=1}^{m} \Pr[f_l(r) = 0] = q^{-m} \ .$$

Now we can calculate the expected value of $|V(G, f_1, \ldots, f_m)|$ as follows.

$$\begin{aligned} E[|V(G, f_1, \ldots, f_m)|] &= \sum_{r \in V(G)} \Pr[r \in V(f_1, \ldots, f_m)] \cdot 1 \\ &= \sum_{r \in V(G) \setminus \{s\}} \Pr[r \in V(f_1, \ldots, f_m)] + \Pr[s \in V(f_1, \ldots, f_m)] \\ &= \sum_{r \in V(G) \setminus \{s\}} q^{-m} + 1 = \frac{|V(G)| - 1}{q^m} + 1 \end{aligned}$$

We calculated, that the expected number of candidates for the secret key s is $\frac{|V(G)|-1}{q^m} + 1$. This number depends on $|V(G)|$ and m. As we have seen in Corollary 39, the size of $V(G)$ is at most $(\nu + 1)^n$. We will try to optimize the actual value to reach this bound. m, the maximum number of messages, allowed to be encrypted with the same key, is a parameter, we can set according to the desired security level.

7.3 Parameter settings

Let $c \in \mathbb{F}[x_1, \ldots, x_n]$ be a ciphertext, $\mathrm{Dec}_{SK}(c) = c(s) = m$. Suppose there is an attacker, who has the public multiplicative key $MK = G$ and a set $\{f_1, \ldots, f_m\} \subset S$, i.e. $f_l(s) = 0,\ l = 1, \ldots, m$. Let him perform a KPA/CPA attack using the Algorithm 7 and let V be the output of Algorithm 7, $V = \{r^{(1)}, \ldots, r^{(|V|)}\}$, $s \in V$. The attacker has gained the following information about $Dec_{SK}(c) = m$:

$$m \in c(V) = \{c(r) \mid r \in V\} \ .$$

We would like to calculate the probability, that an attacker has gained no information about the plaintext m, by computing V. He knows, that a plaintext m is in $c(V) = \{c(r) \mid r \in V\}$. If $c(V) = \mathbb{F}$, then he has no information. Here we need to point out, that in fact, it might happen, that even though $c(V) = \mathbb{F}$, there would exist such $a, b \in \mathbb{F}$, that

$$|\{r \in V \mid c(r) = a\}| \gg |\{r \in V \mid c(r) = b\}|$$

Then $m = a$ would be likelier than $m = b$ and the attacker would have gained some information. However, the values of $c(r)$ are distributed uniformly over $\mathbb{F}$, because we have $c = f+m$, for some random $f \in S$, $r \in V(G)$, hence by Corollary 48, the values of $f(r)$ are distributed uniformly over $\mathbb{F}$. The addition of m only shifts all the values by a constant. This does not affect the uniform distribution, hence $c(r)$ are distributed uniformly over $\mathbb{F}$. We conclude that the scenario, where many values of $c(r)$ accumulate in one point, is unlikely and we further suppose, that if $c(V) = \mathbb{F}$, then the attacker has gained no information.

What is $\Pr[c\,(V) = \mathbb{F}]$? We have shown, that $c(r^{(1)}), \ldots, c(r^{(|V|)})$ are independent values, distributed uniformly over $\mathbb{F}$, i.e. for a fixed $a \in \mathbb{F}$ and $r \in V$, we have $\Pr[c\,(r) = a] = \frac{1}{|\mathbb{F}|} = \frac{1}{q}$. We have

$$\Pr[a \notin c\,(V)] = \Pr[c\,(r) \neq a, \forall r \in V] = \prod_{i=1}^{|V|} \Pr[c(r^{(i)}) \neq a] = \left(\frac{q-1}{q}\right)^{|V|} .$$

We denote $\{a_1, \ldots, a_q\} = \mathbb{F}$. Then, by inductive application of the Law of total probability, we get

$$\begin{aligned}\Pr[c\,(V) = \mathbb{F}] &= 1 - \Pr[a_1 \notin c\,(V)] - \Pr[a_2 \notin c\,(V)\,,\ a_1 \in c\,(V)] - \ldots - \\ &\quad - \Pr[a_q \notin c\,(V)\,,\ a_1, \ldots, a_{q-1} \in c\,(V)] \\ &> 1 - \Pr[a_1 \notin c\,(V)] - \Pr[a_2 \notin c\,(V)] - \ldots - \Pr[a_q \notin c\,(V)] \\ &= 1 - \sum_{k=1}^{q} \Pr[a_k \notin c\,(V)] = 1 - q \cdot \left(\frac{q-1}{q}\right)^{|V|} .\end{aligned}$$

Let us assume, that $|V| \gg q$. We set

$$b := \frac{|V|}{q} .$$

Then

$$\Pr[c\,(V) = \mathbb{F}] > 1 - q \cdot \left(\left(1 + \frac{-1}{q}\right)^{q}\right)^{b} \approx 1 - \frac{q}{e^b} .$$

If we further assume, that the size of the output V is its expected value, calculated in the previous section,

$$\begin{aligned}|V| = E[|V|] &= E[|V\,(G, f_1, \ldots, f_m)\,|] \\ &= \frac{|V\,(G)\,| - 1}{q^m} + 1\end{aligned}$$

then we get, that the chance, the attacker has gained no information about the plaintext is

$$\Pr[c\,(V) = \mathbb{F}] > 1 - \frac{q}{e^b} \ , \ b = \frac{|V\,(G)\,| - 1}{q^{m+1}} + \frac{1}{q} .$$

Table 7.1 illustrates approximate values of $1 - \frac{q}{e^b}$ for different choices of parameters q, n, ν, m, when we set $|V(G)| := (\nu + 1)^n$, the optimistic value. Notice, that the

Table 7.1: Lower bounds on $\Pr[c(V) = \mathbb{F}]$

		$m = 6$	$m = 7$	$m = 8$	$m = 9, \ldots, 15$	$m = 16$	$m = 17$
$q = 9$	$n = 10$	1	0,99997	< 0	< 0	< 0	< 0
$\nu = 5$	$n = 20$	1	1	1	1	< 0	< 0
$q = 81$	$n = 10$	1	1	0,94282	< 0	< 0	< 0
$\nu = 40$	$n = 20$	1	1	1	1	1	< 0
$q = 2^{10}$	$n = 10$	1	1	1	< 0	< 0	< 0
$\nu = 2^9$	$n = 20$	1	1	1	1	1	1

table does not give the values of $\Pr[c(V) = \mathbb{F}]$. It only shows its approximate lower bounds.

We notice, that for a large m, the value of $1 - \frac{q}{e^b}$ is less than 0. This happens when $|V(G)| - 1 < q^{m+1}$ and the presumed $|V| \gg q$ does not hold. In such cases, the bound is not very accurate. We interpret is as it does not give us any information on $\Pr[c(V) = \mathbb{F}]$. However, we can guess, that the probability is very low. In fact it reaches zero once we get to $|V| < q$.

For small m, the table says, that the value of $1 - \frac{q}{e^b}$ is 1. Apparently, the actual value of $1 - \frac{q}{e^b}$ must be less than 1, as $\frac{q}{e^b}$ is a positive number. The value given by the table is only approximate and we interpret it as $\Pr[c(V) = \mathbb{F}]$ is very high.

Notice, that for fixed q, ν, n there is such $\widetilde{m} \in \mathbb{N}$, that for $m = 1, \ldots, \widetilde{m}$, the $\Pr[c(V) = \mathbb{F}]$ is close to 1, i.e. the cryptosystem is secure against the attack by Algorithm 7 and for $m > \widetilde{m}$, the $\Pr[c(V) = \mathbb{F}]$ may be very low and the attack might have a high chance of a success. In the table we can also see, that the rate of $\frac{\widetilde{m}}{n}$ ranges between 0,70 and 0,85 and it grows with q and n. The table does not show this information, but we can calculate, that for $q = 2^{32}, \nu = 2^{31}, n = 20$ the rate is 0,95 and $\widetilde{m} = 19 = n - 1$.

We conclude, that for $q = 2^{32}, \nu = 2^{31}$ and higher, the setting of the parameter $m = n - 1$ is secure against the attacks described in Chapter 6. We need to emphasize here, that this security is only achieved, if the optimistic condition $|V(G)| = (\nu + 1)^n$ holds. In the following chapter we show, how we can ensure that this condition holds.

8. Symmetric Polly Cracker - version 2

In Chapters 6 and 7 we have seen, that the security of SymPC1 increases with the size of the algebraic set $V(G)$. In this chapter we present a second version of SymPC, called SymPC2, where we intend to maximize $V(G)$. As a co-product of this adjustment we get a lower complexity of the Setup phase of the algorithm.

The new version SymPC2 is described by Algorithm 8. It uses the same set of messages $\mathcal{P} = \mathbb{F}$, the set of ciphertexts is $\mathcal{C} \subset \mathbb{F}[x_1, \ldots, x_n]$ and again the keys $K \in \mathcal{K}$ are pairs (SK, MK), where the secret key $SK = s$ is a vector in $\mathbb{F}^n$ and $MK = \{g_1, \ldots, g_n\}$, the multiplication key, is a set of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. In addition, there is a new parameter $\xi \in \mathbb{N}$, which says "how sparse" are the polynomials on the output of the function Encrypt.

The Algorithm 8 differs from Algorithm 5 in the phases Setup and Encrypt. In Setup, the polynomials g_i in the multiplicative key are in $\mathbb{F}[x_i]$ and they decompose to linear factors over $\mathbb{F}$. Furthermore, they are square-free. As we will show in Propositions 51 and 52, such choice maximizes the size of $V(G)$.

In function Encrypt, the polynomials f are chosen in such way, that the number of non-zero coefficients of $f \in \mathcal{L} = \mathbb{F}[x_1, \ldots, x_n]/\langle g_1, \ldots, g_n \rangle$ is at most $\xi - 1$ and therefore the number of non-zero coefficients of the ciphertext is at most ξ. If $\xi \ll (\nu+1)^n$, we get that the fresh encryptions are sparse polynomials in $\mathcal{L}$. This should speed up the evaluation performed in functions Encrypt and Decrypt and also multiplication and addition of fresh ciphertexts. Note, that if a ciphertext $c \in \mathcal{L}$ has been obtained by multiplying a large number of fresh encryptions, it no longer needs to be sparse, therefore the complexity of Decrypt does not get improved in general, only in the case of decrypting fresh encryptions.

Further in this chapter we conjecture, that these modifications do not affect the security in any way, other than the mentioned improvement, caused by optimizing the size of $V(G)$.

Proposition 49. *Algorithm 8 describes a cryptosystem, that is additively and multiplicatively homomorphic.*

Proof. Algorithm 8 describes a cryptosystem, that is a variant of cryptosystem SymPC1. □

Proposition 50. $\{g_1, ..., g_n\}$ *is a reduced Gröbner basis of* $\langle g_1, ..., g_n \rangle$*, the polynomials in* $\mathbb{F}[x_1, ..., x_n]/\langle g_1, ..., g_n \rangle$ *have a degree at most* $\nu \cdot n$*,* $\langle g_1, ..., g_n \rangle$ *is a zero-dimensional ideal and*

$$\dim_{\mathbb{F}} \left(\mathbb{F}[x_1, ..., x_n]/\langle g_1, ..., g_n \rangle \right) = (\nu + 1)^n \quad .$$

Proof. The polynomials $\{g_1, ..., g_n\}$, chosen in the Setup of Algorithm 8 are of the form of polynomials chosen in the Setup of the Algorithm 5, hence the proof is a consequence of Propositions 37 and 38. □

Proposition 51. $|V(g_1, ..., g_n)| = (\nu + 1)^n$

Algorithm 8 SymPC2

SETUP

Input: $n, \nu, q \in \mathbb{N}, \nu < q-1,\ q$ is a prime power

Output: $SK \in \mathbb{F}^n, MK \subset \mathbb{F}[x_1, \ldots, x_n]$

set $\mathbb{F} := \mathbb{F}_q$
choose $s = (s_1, ..., s_n) \leftarrow \mathbb{F}^n$
for $i = 1 \rightarrow n$ **do**
  for $l = 1 \rightarrow \nu$ **do**
    choose $t_l^{(i)} \leftarrow \mathbb{F} \setminus \left\{t_1^{(i)}, \ldots, t_{l-1}^{(i)}\right\}$
  end for
end for
for $i = 1 \rightarrow n$ **do**
  set $g_i := (x_i - s_i) \prod_{l=1}^{\nu} \left(x_i - t_l^{(i)}\right)$
end for
set the secret key $SK := s$
set the multiplication key $MK := \{g_1, \ldots, g_n\}$
return (SK, MK)

ENCRYPT$_\xi$

Input: message $m \in \mathbb{F}, SK \in \mathbb{F}^n, \nu \in \mathbb{N}$

Output: $c \in \mathcal{L}$

set $f := 0 \in \mathbb{F}[x_1, \ldots, x_n]$
for $l = 1 \rightarrow \xi - 1$ **do**
  choose $a \leftarrow \mathbb{F}$
  choose $(i_1, \ldots, i_n) \leftarrow \{0, \ldots, \nu\}^n$
  set $f := f + a \cdot x_1^{i_1} \cdots x_n^{i_n}$
end for
set $c = f - f(SK) + m \in \mathbb{F}[x_1, \ldots, x_n]/\langle g_1, \ldots, g_n \rangle$
return c

DECRYPT

Input: $c \in \mathcal{L},\ SK \in \mathbb{F}^n$

Output: $m \in \mathbb{F}$

set $m := c(SK)$
return m

ADD

Input: $c_1, c_2 \in \mathcal{L}$

Output: $c \in \mathcal{L}$

set $c := c_1 + c_2 \in \mathcal{L}$
return c

MULT$_{MK}$

Input: $c_1, c_2 \in \mathcal{L}$

Output: $c \in \mathcal{L}$

set $c := c_1 \cdot c_2$
for $i = 1 \rightarrow n$ **do**
  set $c := c \mod g_i$
end for
return c

Proof.

$$V(g_1, ..., g_n) = \left\{ r = (r_1, \ldots, r_n) \in \mathbb{F}^n \mid r_i \in \left\{ t_1^{(i)}, \ldots, t_\nu^{(i)}, s_i \right\}, i = 1, \ldots, n \right\} \ .$$

Apparently, there are $(\nu+1)^n$ different choices of $r \in V(g_1, ..., g_n)$. □

Proposition 52. *For given parameters ν and n, the choice of the set $G = \{g_1, ..., g_n\}$ in the Setup phase of SymPC2 maximizes the size of $V(G)$.*

Proof. As we have seen in Proposition 29, the size of an algebraic set of a zero-dimensional ideal I in $\mathbb{F}[x_1, \ldots, x_n]$ is at most $\dim_{\mathbb{F}}(\mathbb{F}[x_1, \ldots, x_n]/I)$. We have

$$\dim_{\mathbb{F}}(\mathbb{F}[x_1, \ldots, x_n]/G) = (\nu+1)^n = V(G) \ .$$

□

8.1 Complexity of SymPC2

In this section we investigate the impact of the modification of functions Setup and Encrypt on the complexity of each function of the algorithm SymPC2.

Setup The generation of polynomials $\{g_1, \ldots, g_n\}$ has become a lot simpler. For each g_i we need to perform ν multiplications of polynomials of degrees at most ν and one in $\mathbb{F}[x_i]$. The complexity is $O(n \cdot \nu^2)$ operations in $\mathbb{F}$.

Encrypt Again, the most time-consuming operation is the evaluation of the polynomial f in the point $s \in \mathbb{F}^n$. This can be divided into evaluation of monomials of f and later addition of these. We have ξ monomials, each evaluation consists of at most $\nu \cdot n$ multiplications in $\mathbb{F}$. Overall we get the complexity of $O(\xi \cdot \nu \cdot n)$ operations in $\mathbb{F}$.

Decrypt The complexity of this function strongly depends on the nature of the ciphertext on the input. If it is a fresh encryption, i.e. it is a sparse polynomial, then the complexity is the same as the complexity of Encrypt in SymPC2, that is $O(\xi \cdot \nu \cdot n)$ operations in $\mathbb{F}$. If it is a ciphertext, that has been created by evaluation of ciphertext polynomial of a high degree or of a ciphertext polynomial, that has many monomials, i.e. the ciphertext is no longer sparse, then the complexity of Decrypt is the same as in SymPC1, that is $O(n \cdot (\nu+1)^n)$.

Add Similarly, the complexity of the function Add ranges from $O(\xi)$ operations in $\mathbb{F}$ for a fresh sparse ciphertext to $O((\nu+1)^n)$ operations in $\mathbb{F}$ for a dense composite ciphertext.

Mult The complexity of the first part of the function - multiplication ranges from $O(\xi^2)$ to $O((\nu+1)^{2n})$ operations in $\mathbb{F}$, analogically to the function Add. The later part - reduction performs at most n reductions modulo a polynomial in $\mathbb{F}[x_i]$. For a dense ciphertext, each reduction by g_i of degree $\nu+1$ takes $O((\nu+1)((\nu+1)^n - (\nu+1)))$ operations in $\mathbb{F}$, overall that is $O(n \cdot (\nu+1)^{n+1})$ operations. When we have a sparse polynomial and reduce it, it becomes dense quite fast. We see, that for sparse ciphertexts we get complexity $O(n \cdot (\nu+1)^{n+1})$ and for dense ciphertexts we get $O((\nu+1)^{2n})$ operations in $\mathbb{F}$.

Table 8.1: Complexity of SymPC1 and SymPC2

	SymPC1	SymPC2	
		sparse	dense
Setup	$\frac{\nu^n}{n!}$	$n \cdot \nu^2$	$n \cdot \nu^2$
Encrypt	$n \cdot (\nu+1)^n$	$\xi \cdot \nu \cdot n$	$\xi \cdot \nu \cdot n$
Decrypt	$n \cdot (\nu+1)^n$	$\xi \cdot \nu \cdot n$	$n \cdot (\nu+1)^n$
Add	$(\nu+1)^n$	ξ	$(\nu+1)^n$
Mult	$(\nu+1)^{2n}$	$n \cdot (\nu+1)^n$	$(\nu+1)^{2n}$

The Table 8.1 shows the comparison of the complexity of algorithms SymPC1 and SymPC2. We omit the O-notation.

8.2 Security of SymPC2

What are the security implications of the changes, we made? We have maximized the size of $V(G)$, which made it possible to count on the lower bounds from Table 7.1. In other words, if we encrypt no more than about $n-1$ messages with one key, the cryptosystem is secure against the two attacks from Chapter 6.

Let us have a look at the complexity implications on the attack by computing algebraic sets (described by Algorithm 7). The phase where the algorithm calculates the algebraic set $V(G)$ has become simpler as the attacker only checks evaluations in $n \cdot \nu$ points. On the other hand, the initial size of $V(G)$ is larger, so it takes more calculations to find $V(G, f_1), V(G, f_1, f_2)$, etc.

Again, no matter what the complexity of the attack is, we can prevent the attack from succeeding the same way as before.

In Section 7.1 we showed, that the distribution of zeros of polynomials from S/G is uniformly random. We would like to be able to prove, that the same holds in the case of sparse polynomials. We do not see a reason, why a polynomial should have more zeros or less zeros or why the zeros should accumulate somewhere just because we work with a sparse polynomial, but we were unable to find a formal proof.

Conjecture 53. *Choose $f \in S/G$ sparse as in the function Encrypt of SymPC2. Then $\varphi(f)$ is distributed uniformly over $\mathbb{F}^{t-1} \times \{0\}$. In particular, for $r \in V(G)$, $r \neq s$ it holds*

$$\Pr[f(r) = 0] = \frac{1}{|\mathbb{F}|} = q^{-1} \ .$$

Conclusion

In this thesis we studied the alternative approaches to fully homomorphic encryption, i.e. approaches, that are not lattice-based. First we described the scheme by Arkmecht et al., whose security is based on the hardness of decoding in a random linear code. Then we have looked at the family of cryptosystems called Polly Cracker and studied its limitations. The main contribution of this thesis is a design of a new symmetric fully homomorphic encryption scheme in which we applied the new idea of using Gröbner bases as a tool for decreasing the size of ciphertexts, therefore decreasing the complexity of all the operations on the set of ciphertexts.

At the very end, let us make a rough comparison of the mentioned homomorphic encryption schemes. As we suggested, all the homomorphic encryption schemes, that have been developed so far, have some issues, that make them impractical. The Table 8.2 shows the schemes mentioned in the thesis (and one other scheme) and their particular limitations and disadvantages. First, let us make a short informal description of the studied limitations.

μ-bound: There is a bound on the number of multiplications performed on ciphertexts. In other words, if we have fresh encryptions $c_1, \ldots, c_k \in \mathcal{C}$, we are only allowed to evaluate polynomials $h \in \mathcal{C}[y_1, \ldots, y_k]$ of degree at most μ. Evaluations of polynomials of greater degree do not necessarily decrypt correctly.

m-bound: There is a bound on the number of messages allowed to be encrypted with the same key. This bound is imposed, because there exists a KPA or CPA attack, where the attacker collects $m+1$ independent pairs of plaintext-ciphertext and breaks the cryptosystem. By independent, we mean that there exists no relation h, such that $c_{m+1} = h(c_1, \ldots, c_m)$, $h \in \mathcal{C}[y_1, \ldots, y_m]$. The non-existence of as many pairs obviously prevents the attacker from gaining them. This limitation is only relevant to symmetric cryptosystems. In public-key cryptosystems, the attacker can create himself as many plaintext-ciphertext pairs as he likes, as the encryption key is public.

divergence: When we multiply two ciphertexts, the size of the resulting ciphertext is greater than the two previous, i.e. if $c_1, \ldots, c_k \in \mathcal{C}$, then the length of $c = h(c_1, \ldots, c_k)$, $h \in \mathcal{C}[y_1, \ldots, y_k]$ grows with the degree of h. There is either no bound on the size of a $c \in \mathcal{C}$ or there is one, which is so high, that the implications on the complexity of decryption and other operations make the scheme impractical.

complexity: Informally, if all the parameters of the scheme are set up in such way, that the cryptosystem is secure against the identified threats, the complexity of either key generation, encryption, decryption or operations with ciphertexts make the scheme impractical. Description of this limitation is very informal. We advise the reader to take it into account.

Table 8.2: Comparison of homomorphic encryption schemes

	type	μ-bound	m-bound	divergence	complexity
Polly Cracker	public-key			•	•
Gentry	public-key				•
Armknecht	symmetric	•	•		
SymPC1	symmetric		•		•
SymPC2	symmetric		•		

The schemes we compare are the public-key Polly Cracker, where we consider the variant described in Chapter 4, Craig Gentry's Fully Homomorphic Encryption Scheme [Gen09] mentioned in the Introduction, which is also public-key, the symmetric Code-based Encryption Scheme proposed by Armknecht et. al. [AAPS11] and described in Chapter 2 and the two versions of our new scheme SymPC.

List of Algorithms

List of Tables

Bibliography

[AAPS11] Frederik Armknecht, Daniel Augot, Ludovic Perret, and Ahmad-Reza Sadeghi. On constructing homomorphic encryption schemes from coding theory. *IACR Cryptology ePrint Archive*, 2011:309, 2011.

[BCD+08] Peter Bogetoft, Dan Lund Christensen, Ivan Damgard, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, and Tomas Toft. Multiparty computation goes live. Cryptology ePrint Archive, Report 2008/068, 2008. `http://eprint.iacr.org/`.

[CLO97] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997.

[FK94] M. Fellows and N. Koblitz. Combinatorial cryptosystems galore! In G. L. Mullen and P. J.-S. Shiue, editors, *Finite Fields: Theory, Applications, and Algorithms*, volume 168 of *Contemporary Mathematics*, pages 51–61. 1994.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1992.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.

[SBff11] D. Stanovský, L. Barto, and Univerzita Karlova. Matematicko fyzikální fakulta. *Počítačová algebra*. Matfyzpress, 2011.

[Sti95] Douglas R. Stinson. *Cryptography - theory and practice*. Discrete mathematics and its applications series. CRC Press, 1995.

[Win96] F. Winkler. *Polynomial Algorithms in Computer Algebra*. Texts and Monographs in Symbolic Computation. Springer, 1996.